

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Г. М. Цибульский

подпись

« ____ » _____ 2017 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 — «Информационные системы и технологии»

Разработка информационной системы с web-интерфейсом для управляющей
компании ООО «Алиса» г. Красноярск

Руководитель _____
подпись, дата

ст. преподаватель каф. СИИ А. В. Пятаева

Выпускник _____
подпись, дата

Ю. С. Черненко

Консультант

подпись, дата

Д. А. Перфильев

Нормоконтролер

подпись, дата

М. А. Аникьева

Красноярск 2017

Продолжение титульного листа бакалаврской работы по теме
«Разработка информационной системы с web-интерфейсом для управляющей
компании ООО «Алиса» г. Красноярск».

Нормоконтролер

подпись, дата

М. А. Аникьева

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка информационной системы с web-интерфейсом для управляющей компании ООО «Алиса» г. Красноярск» содержит 62 страницы текстового документа, 2 приложения, 19 использованных источников, 24 рисунка.

ИНФОРМАЦИОННАЯ СИСТЕМА, УПРАВЛЯЮЩАЯ КОМПАНИЯ, УДАЛЕННЫЙ ДОСТУП, MYSQL, Yii2, MVC, PHP.

Цель работы — разработка информационной системы с web-интерфейсом для управляющей компании ООО «Алиса» г. Красноярск.

Для достижения поставленной цели были сформулированы следующие задачи:

- а) анализ предметной области;
- б) проектирование информационной системы с web-интерфейсом;
- в) разработка информационной системы с web-интерфейсом.

В ходе выполнения работы приведено сравнение с аналогичными программными продуктами. Была спроектирована и разработана информационная система «ИС Алиса» на языке программирования PHP. Также было проведено тестирование разработанной информационной системы на предприятии.

СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области	7
1.1 Анализ работы управляющей компании.....	7
1.2 Общие сведения о существующих информационных системах в сфере жилищно-коммунального хозяйства.....	8
1.3 Выводы по главе 1.....	12
2 Проектирование информационной системы	14
2.1 Анализ средств проектирования.....	14
2.2 Анализ программной платформы.....	16
2.3 Диаграмма вариантов использования	17
2.4 Диаграмма компонентов проектируемой информационной системы.....	21
2.5 Диаграмма развертывания проектируемой информационной системы....	22
2.6 Диаграмма сущность-связь проектируемой информационной системы ..	23
2.7 Диаграмма последовательности проектируемой информационной системы	25
2.8 Проектирование дизайна системы	27
2.9 Выводы по главе 2.....	28
3 Разработка информационной системы с web-интерфейсом	29
3.1 Настройка ролевого управления доступом	29
3.2 Панель управления сотрудника	33
3.3 Интерфейсы	34
3.4 Выводы по главе 3.....	40
Заключение	42
Список использованных источников	43
Приложение А Техническое задание	45
Приложение Б Плакаты презентации.....	54

ВВЕДЕНИЕ

Жилищно-коммунальное хозяйство (в дальнейшем ЖКХ) представляет собой отрасль сферы услуг и важнейшую часть территориальной инфраструктуры, определяющую условия жизнедеятельности человека, прежде всего комфортность жилища, его инженерное благоустройство, бытовые и другие услуги, от которых зависит состояние здоровья и качество жизни.

Для непрерывного и эффективного функционирования работы управляющей компании необходимо иметь правильно построенную систему работы с данными.

Использование информационных систем и технологий позволяет решить ряд следующих задач: поиск, обработка и хранение данных, исследование способов представления и хранения информации, анализ и прогнозирование потоков информации различных видов и типов. А также удаленный доступ к необходимой информации и систематизация различных данных, так как по мере своей работы специалисту, как правило, приходится использовать определенные справочники, нормативно-правовую базу, инструктивно-распорядительную документацию, подкрепляя это своими знаниями и навыками.

Большая часть российских управляющих компаний использует либо индивидуально разработанные корпоративные информационные системы, либо программное обеспечение пакета «Microsoft Office». Также список программного обеспечения можно пополнить следующими продуктами: «1С: Учет в управляющих компаниях ЖКХ, ТСЖ и ЖСК», «Microsoft Dynamics CRM», «АИС Жилищно-коммунальное хозяйство».

Однако в ходе анализа функциональных возможностей был выявлен ряд значительных недостатков в упомянутых программных продуктах.

Следовательно, необходимость разработки информационной системы, предназначенной для работы с данными, ориентированной на сотрудника управляющей компании и ее клиентов, очевидна. В результате разработки предполагается получить web-портал удаленного доступа, включающий в себя

базы данных и личный кабинет для сотрудника и для клиентов, обслуживаемых компанией «Алиса».

Целью выпускной квалификационной работы является разработка информационной системы с web-интерфейсом для управляющей компании ООО «Алиса», г. Красноярск.

Для получения результата необходимо решить следующие задачи:

- а) анализ предметной области;
- б) проектирование информационной системы;
- в) разработка информационной системы с web-интерфейсом.

1 Анализ предметной области

К предметной области относится функционирование управляющей компании, должностные обязанности специалистов компании, а также, используемые ими на данный момент информационные системы.

1.1 Анализ работы управляющей компании

Управляющая компания ЖКХ осуществляет управление многоквартирным домом, обеспечения его необходимыми ресурсами, поддерживающими надлежащее санитарное и техническое состояние объекта.

Деятельность управляющей компании ЖКХ регулируется жилищным законодательством Российской Федерации. Управляющие компании ЖКХ создаются с целью обеспечения жизнедеятельности многоквартирных домов, и все свои услуги они осуществляют только в соответствии с установленными требованиями законодательства.

Как и у любой организации в гражданских правоотношениях, у управляющей компании ЖКХ есть свои определённые обязанности перед собственниками жилья. Например, такие компании обязаны проводить следующие работы:

- поддерживать техническое состояние дома в надлежащем виде, делая его безопасным для жильцов;
- проверка состояния вверенного имущества раз в полгода: крыша, чердак, подвал, системы водоотведения и канализации и так далее. Все обнаруженные проблемы должны документироваться и устраняться в разумные сроки;
- выполнение мелкого и текущего ремонта зданий, и сооружений;
- для проведения капитальных ремонтных работ необходимо получить согласие собственников жилья;
- подготовка дома и прилегающей территории к зиме (весне);

- в случае возникновения неполадок и аварий в доме, при устранении таких проблем необходимо руководствоваться установленными техническими рекомендациями и регламентами;
- создание благоприятных условий для предоставления гражданам коммунальных услуг соответствующего качества;
- контроль за соблюдением норм и правил пожарной безопасности;
- регистрация граждан в квартирах;
- уведомление их о существующих проблемах, общей информации в сфере ЖКХ;
- сбор коммунальных услуг с собственников жилья и информирование их об изменении существующих тарифов [1].

1.2 Общие сведения о существующих информационных системах в сфере жилищно-коммунального хозяйства

На сегодняшний день большая часть российских компаний, управляющих жилищно-коммунальным хозяйством, используют либо индивидуально разработанные корпоративные информационные системы, либо программное обеспечение пакета «MicrosoftOffice». Также, доля предприятий использует программу «1С: Учет в управляющих компаниях ЖКХ, ТСЖ и ЖСК» [2]. Реже используются автоматизированные информационные системы от сторонних разработчиков, например, «АИС Жилищно-коммунальное хозяйство» [3].

В управляющей компании «Алиса» г. Красноярска используется локальная корпоративная информационная система, представленная на рисунке 1. В ней присутствуют все достаточные для работы функции:

- контроль оплаты по каждому абоненту;
- контроль за действиями каждого сотрудника;
- формирование финансовых отчетов, квитанций об оплате;
- ведение базы данных клиентов.

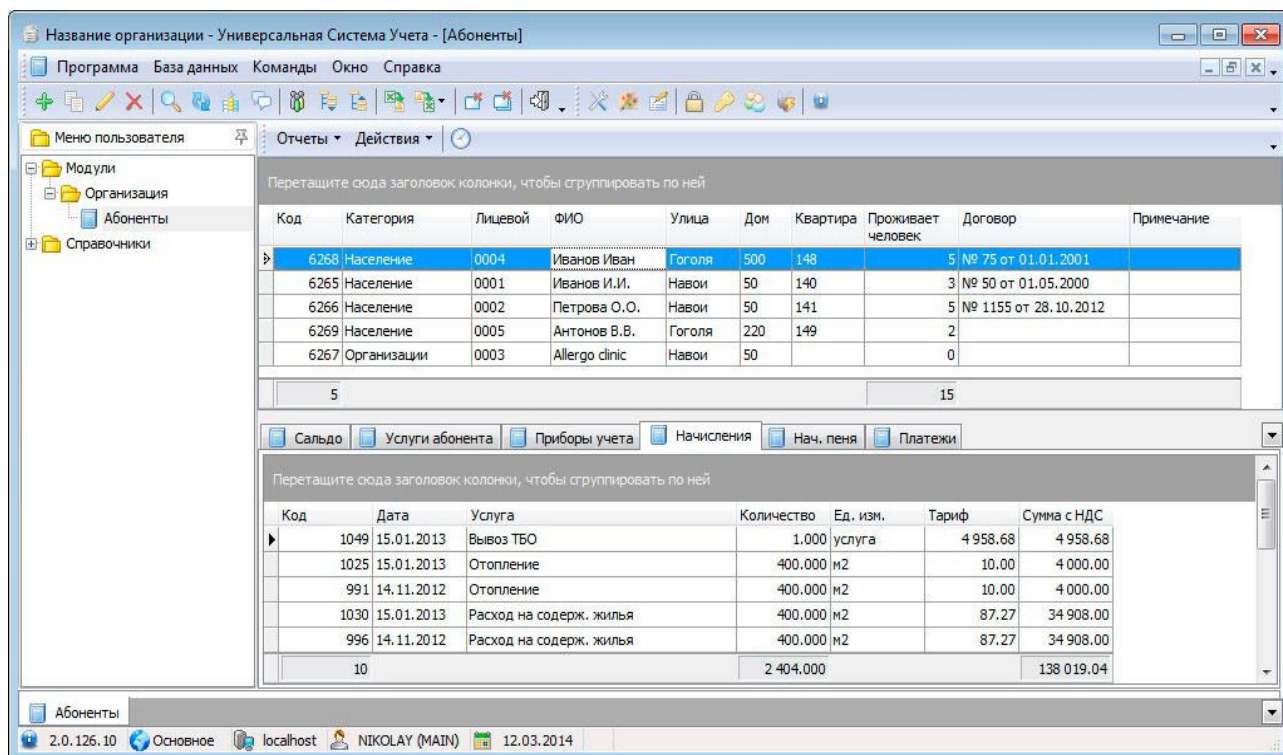


Рисунок 1 — Интерфейс корпоративной информационной системы управляющей компании «Алиса»

С одной стороны, такая информационная система удобна тем, что все хранится в одном месте, и каждый сотрудник может получить доступ к необходимым данным в любое время. Также, для работы в данной системе не требуется особых навыков, можно разобраться в управлении за сравнительно короткое время.

Однако, доступ к любой информации, неограниченный ролевыми рамками, грозит разглашением информации, которая не должна быть доступна для всех сотрудников.

Также, минусом такой информационной системы является то, что она не адаптирована к изменениям, так как имеет заверченный функционал, и для его обновления приходится выпускать новую версию информационной системы и проводить процесс внедрения заново. Это включает в себя, также, обучение персонала. В связи с этим рабочий процесс будет приостановлен на неопределенный временными рамками срок.

К тому же, доступ к информационной системе сотрудник управляющей компании может получать только непосредственно в организации. Это является главным недостатком большинства существующих информационных систем. Так как, учитывая человеческий фактор, можно предположить, что при таком устройстве системы нередко будут случаи непредумышленной порчи данных. Также, в современном мире большинство людей имеют в личном пользовании мобильные устройства, именно поэтому нецелесообразно использовать информационную систему, не имеющую удаленного доступа к базам данных.

Согласно информации разработчиков, программа «1С: Учет в управляющих компаниях ЖКХ, ТСЖ и ЖСК», представленная на рисунке 2, предназначена для автоматизации бухгалтерского и налогового учета, а также для комплексной автоматизации управления бизнесом управляющих компаний и позволяет автоматизировать основные участки управленческого и регламентированного учетов. Бухгалтерский и налоговый учет ведется в соответствии с действующим законодательством Российской Федерации.

Рисунок 2 — Интерфейс программы «1С: Учет в управляющих компаниях ЖКХ, ТСЖ и ЖСК»

В данной системе более детальная проработка функционала, удобный интерфейс, но есть и ряд недостатков. Главным недостатком можно назвать то, что использовать эту систему можно исключительно на рабочем месте. Также, данная программа не подлежит внесению изменений, в связи с этим проблематично подстраивать ее под конкретных пользователей. Наконец, функции не распределены по специалистам и правам доступа. Данный факт повышает вероятность ошибки и потерю информации.

Некоторые управляющие компании города Красноярска уже внедрили информационную систему удаленного доступа, подобную проектируемой. Например, городская управляющая компания «ЖилФонд» [4]. Данная система представлена на рисунке 3.

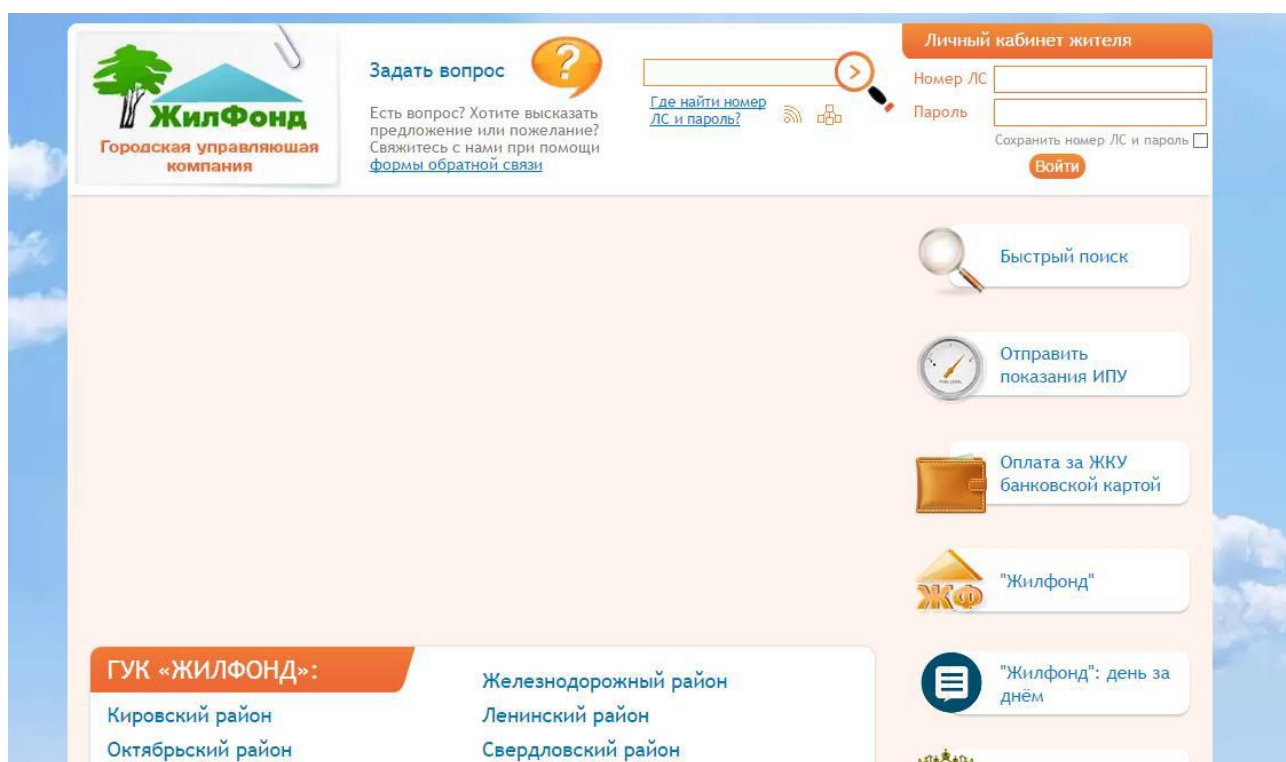


Рисунок 3 — Интерфейс информационной системы для управляющей компании «ЖилФонд»

На сайте представлена вся необходимая информация, как для клиентов, так и для сотрудников данной компании. Также на сайте клиентам предоставлена возможность отправки показаний счетчиков, оплаты жилищно-коммунальных услуг.

В городе Новосибирске используется многофункциональный электронный портал ЖКХ Новосибирской области [5]. Портал изображен на рисунке 4.

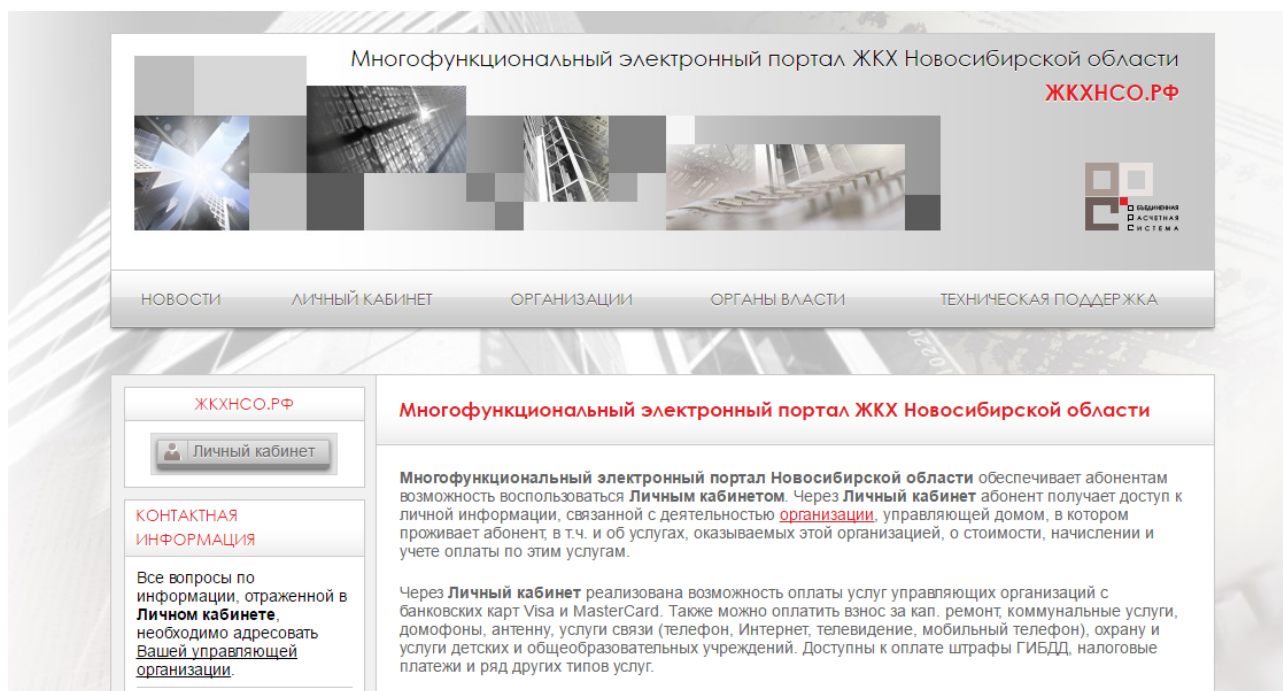


Рисунок 4 — Интерфейс электронного портала ЖКХ в городе Новосибирске

При успешной авторизации для каждого клиента открывается личный кабинет, используя который клиент может:

- получить информацию о балансе лицевого счёта, детализацию начислений по услугам, историю платежей;
- распечатать квитанции и другие отчеты по лицевым счетам;
- отправить показания счётчиков электроэнергии, горячего и холодного водоснабжения.

1.3 Выводы по главе 1

Наиболее значимым недостатком используемых в данный момент информационных систем автоматизации является их неспособность адаптироваться к индивидуальностям и правилам пользователя. Проведение

реорганизации компании, в свою очередь, для приведения ее в соответствие с технологическими требованиями эксплуатации нецелесообразно.

Индивидуально разрабатываемые комплексные информационные системы имеют высокую стоимость и длительный срок разработки и внедрения. Именно поэтому региональным компаниям выгодно привлекать сторонних разработчиков для проектировки информационных систем, не включенных в главную корпоративную информационную систему. Нагрузка на подобные ИС меньше, чем на основную КИС, следовательно, стоимость будет существенно ниже, а время разработки и внедрения — меньше. Значимость подобной работы представляется для управляющих компаний локального значения. Информационное обеспечение сотрудников является важным условием последовательного и динамичного развития компании.

В ходе анализа рабочей деятельности управляющей компании были выявлены необходимые функции, которые обязательно должны присутствовать в разрабатываемой системе. К ним относятся:

- уведомление клиентов о предстоящих событиях;
- отправка заявок на услуги;
- отправка показаний счётчиков.

2 Проектирование информационной системы

Для проектирования информационной системы были выбраны шаблон проектирования MVC, программная платформа Yii2, язык программирования PHP, среда разработки PHP Storm.

2.1 Анализ средств проектирования

Для реализации проекта был выбран шаблон проектирования MVC (Model-View-Controller) — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области [6].

Схема шаблона проектирования MVC представлена на рисунке 5.

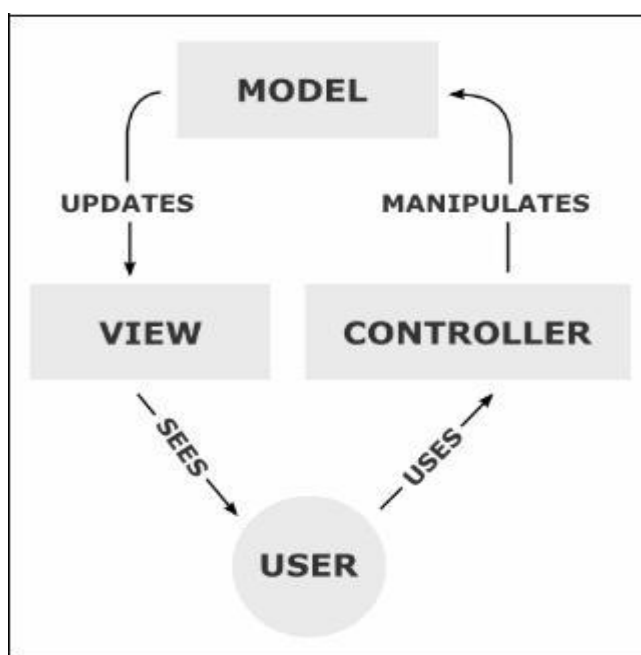


Рисунок 5 — Схема шаблона проектирования Model-View-Controller

Основным назначением шаблона проектирования MVC является разделение бизнес-логики от ее визуализации. Данное разделение предоставляет возможность повторного редактирования проекта, а также его повторного использования [7].

Концепция MVC делит данные, представление и обработку действий пользователя на три компонента:

- модель — представляет знания: данные и методы работы с этими данными, реагирует на запросы посредством изменения своего состояния. Не содержит информации, как визуализировать эти знания;
- представление — отвечает за визуализацию. Например, в качестве представления может выступать форма (окно) с графическими элементами;
- контроллер — обеспечивает связь пользователя и системы. Контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции [8].

Представление и контроллер зависят от модели, однако модель независима от представления и контроллера. Таким образом достигается назначение подобного разделения. Оно позволяет построить независимую модель, а также создать для нее несколько различных представлений.

В качестве системы управления доступом была выбрана RBAC. Управление доступом на основе ролей (англ. RoleBasedAccessControl, RBAC) — развитие политики избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли [9].

Формирование ролей призвано определить четкие и понятные для пользователей компьютерной системы правила разграничения доступа. Ролевое разграничение доступа позволяет реализовать гибкие, изменяющиеся динамически в процессе функционирования компьютерной системы правила разграничения доступа [10].

Такое разграничение доступа является составляющей многих современных компьютерных систем. Как правило, данный подход применяется в системах

защиты СУБД, а отдельные элементы реализуются в сетевых операционных системах. Ролевой подход часто используется в системах, для пользователей которых четко определён круг их должностных полномочий и обязанностей.

2.2 Анализ программной платформы

Фреймворк — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта [11].

Превосходство фреймворка Yii2 над другими программными платформами заключается в эффективности, широких возможностях и качественной документации. Платформа Yii2 изначально спроектирована очень тщательно для соответствия всем требованиям при разработке web-приложений [12].

Платформа Yii2 предоставляет разработчику необходимый инструмент для работы с базами данных — объекты доступа к данным (DAO).

Инструмент Yii DAO — это надстройка над расширением PHP DataObjects (PDO), позволяющая работать с различными СУБД через единый интерфейс. Приложения, разработанные с использованием DAO, могут легко переключаться с одной СУБД на другую без необходимости внесения правок в программный код, отвечающий за доступ к данным.

Шаблон Yii ActiveRecord (AR) реализует переработанный подход объектно-реляционного проектирования (ORM) и ещё больше упрощает работу с базами данных. Представляя таблицу базы данных как класс, а строки таблицы как экземпляры класса, Yii AR избавляет от необходимости написания SQL-выражений, связанных с операциями CRUD (создание, чтение, обновление и удаление) [13].

Yii2 является простым в использовании, а также чрезвычайно гибким и расширяемым фреймворком. Существует много различных сообществ,

поддерживающих пользователей Yii2, ведется активная разработка дополнительных возможностей фреймворка.

Фреймворк Yii2 использует популярный шаблон проектирования Модель-Представление-Контроллер (MVC, Model-View-Controller), который широко применяется в web-программировании.

MVC предназначен для разделения бизнес-логики и пользовательского интерфейса, чтобы разработчики могли легко изменять отдельные части приложения, не затрагивая другие. В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс (например, текст, поля ввода), а контроллер обеспечивает взаимодействие между моделью и представлением.

2.3 Диаграмма вариантов использования

Сценарий использования, вариант использования, прецедент использования (англ. UseCase) — в разработке программного обеспечения и системном проектировании это описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды [14]. Система может отвечать на внешние запросы Актора (англ. Actor), может сама выступать инициатором взаимодействия. Другими словами, сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой, или что система может сделать с «кем» или «чем». Методика сценариев использования применяется для выявления требований к поведению системы, известных также как функциональные требования.

Прецедент описывает некоторый целостный фрагмент поведения системы, не вдаваясь при этом в особенности внутренней структуры субъекта [15]. Определение прецедента содержит все свойственные ему виды поведения: основную последовательность, различные варианты стандартного поведения и различные исключительные ситуации с указанием ответной реакции на них.

В число пользователей проектируемой информационной системы входят:

- сотрудник компании;
- клиент;
- администратор системы.

Сотрудник компании наделен следующими правами:

- просмотр показаний счетчиков;
- экспорт отчета по показаниям счетчиков в формат excel;
- просмотр заявок на услуги от клиентов;
- удаление заявок на услуги;
- добавление новостей на главной странице системы;
- просмотр новостей;
- удаление новостей.

Диаграмма использования для сотрудника изображена на рисунке 6.

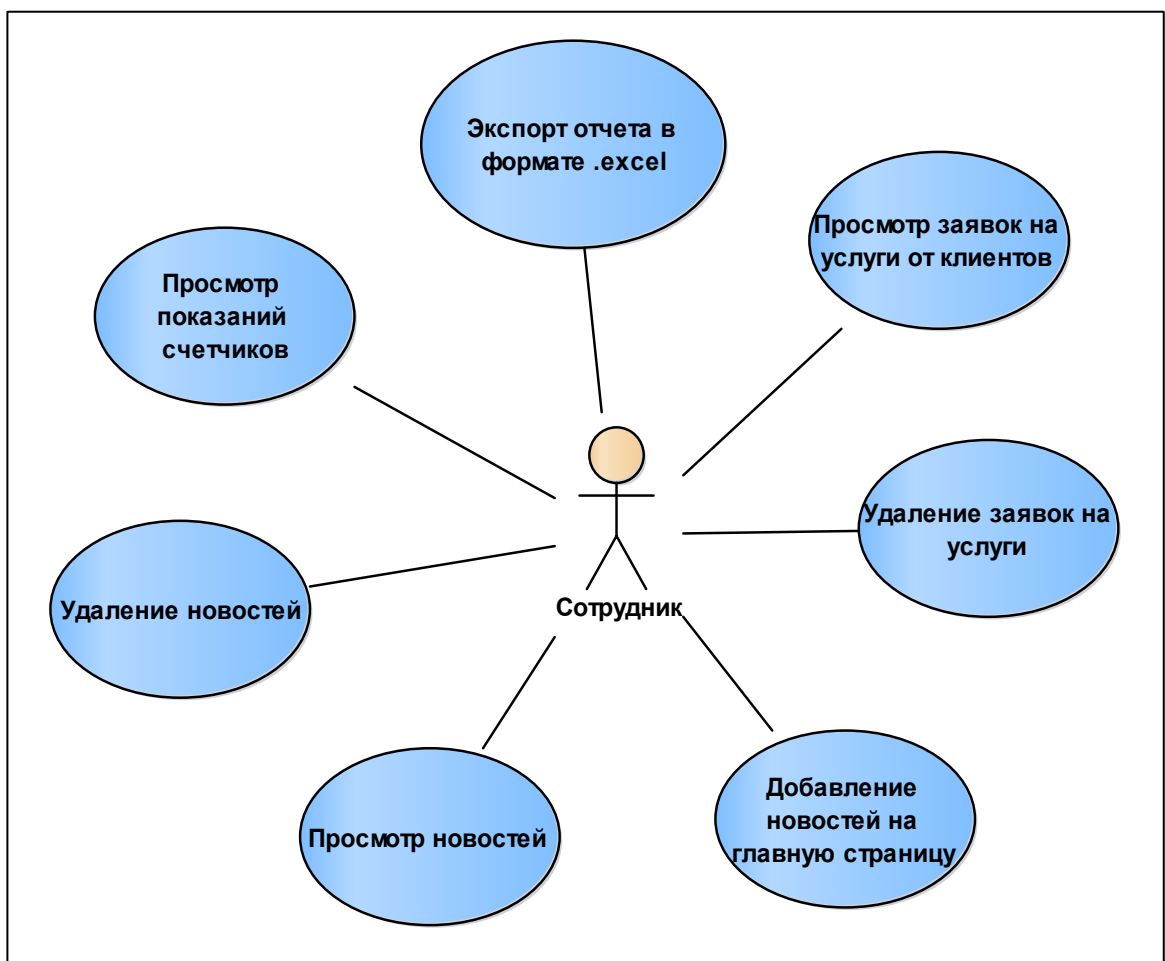


Рисунок 6 — Варианты использования для актора «Сотрудник компании»

Клиент наделен следующими правами:

- отправка показаний счетчиков;
- обновление данных по показаниям счетчиков;
- просмотр данных по показаниям счетчиков;
- отправка заявок на услуги;
- просмотр заявок на услуги;
- просмотр информации в системе.

Диаграмма использования для клиента изображена на рисунке 7.

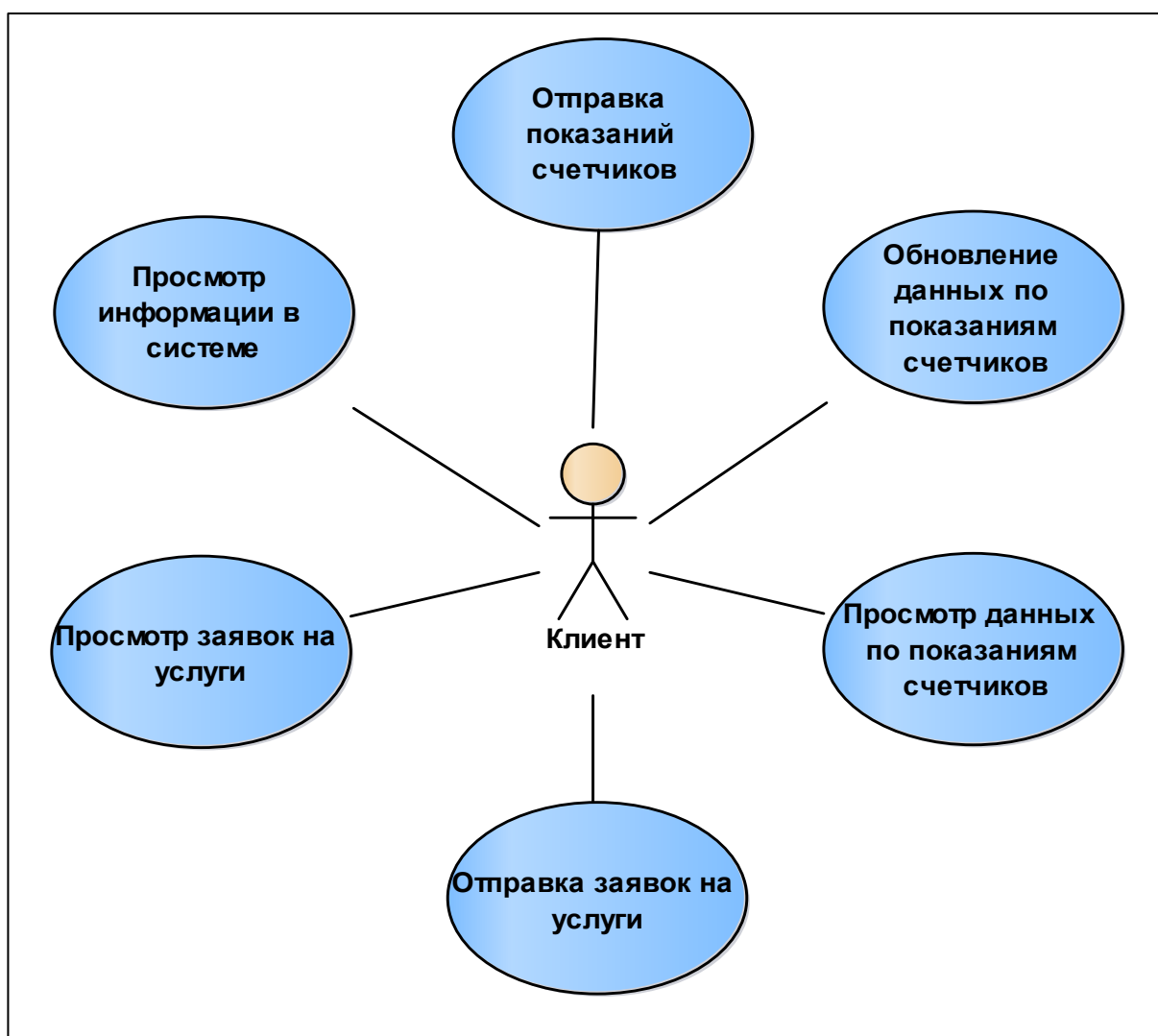


Рисунок 7 — Варианты использования для актора «Клиент»

Администратор системы наделен следующими правами:

- просмотр пользователей ИС;

- создание пользователей ИС;
- обновление пользователей ИС;
- удаление пользователей ИС;
- просмотр ролей пользователей;
- создание ролей пользователей;
- обновление ролей пользователей;
- удаление ролей пользователей.

Диаграмма вариантов использования для администратора системы изображена на рисунке 8.

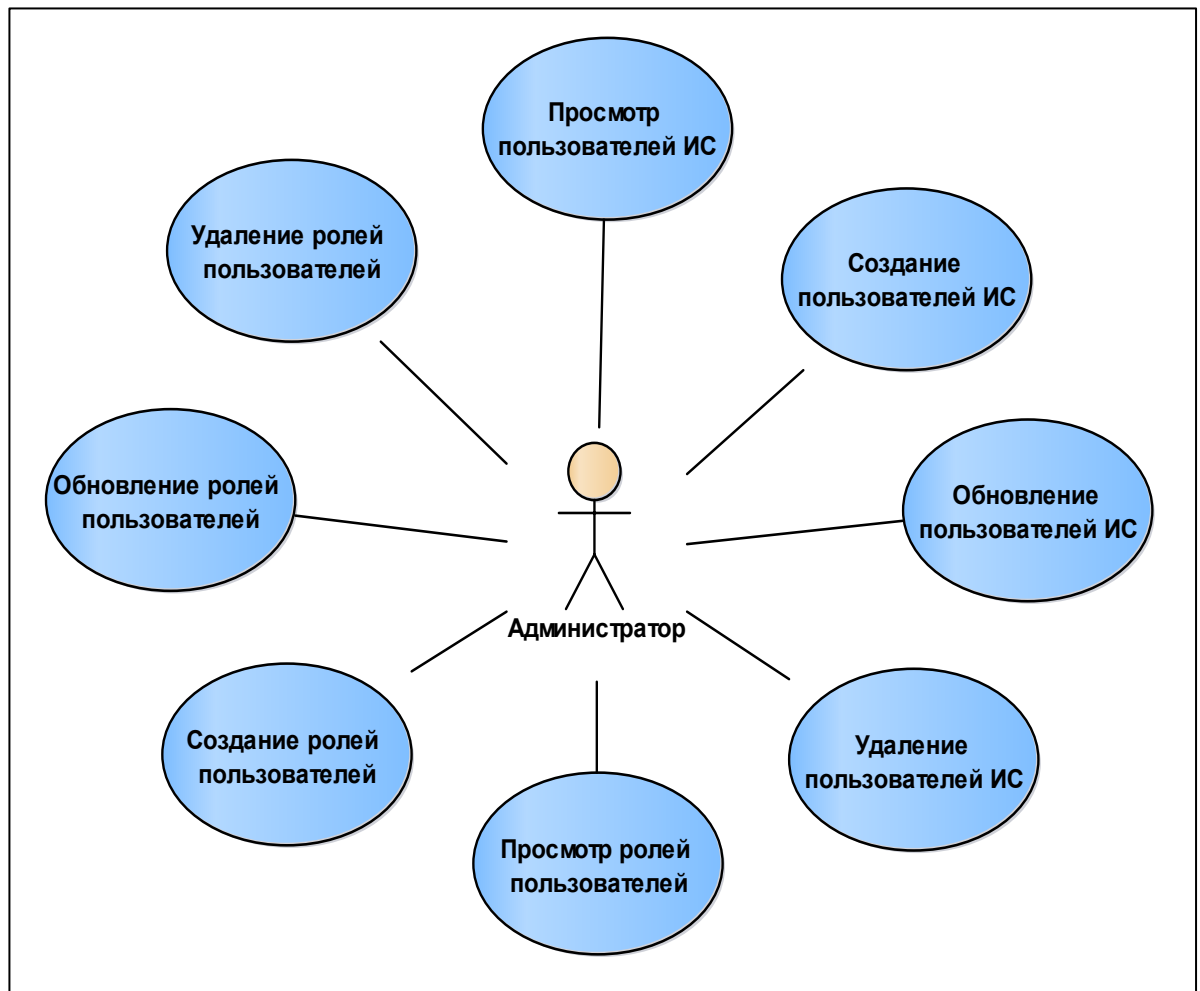


Рисунок 8 — Варианты использования для актора «Администратор системы»

2.4 Диаграмма компонентов проектируемой информационной системы

Диаграмма компонентов — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами [16]. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты. Диаграмма компонентов системы отображена на рисунке 9.

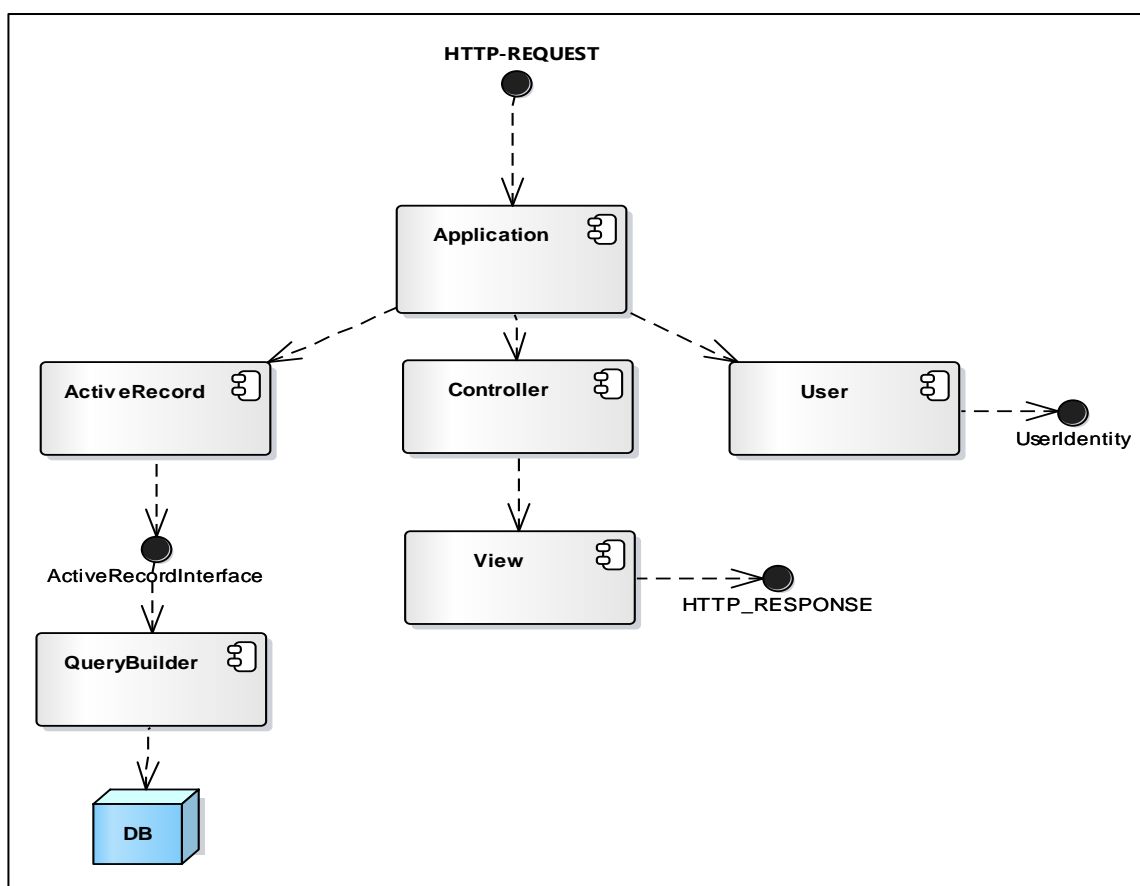


Рисунок 9 — Диаграмма компонентов проектируемой информационной системы

Компоненты связываются через зависимости, когда соединяется требуемый интерфейс одного компонента с имеющимся интерфейсом другого компонента. Таким образом иллюстрируются отношения клиент-источник между двумя компонентами.

Зависимость показывает, что один компонент предоставляет сервис, необходимый другому компоненту. Зависимость изображается стрелкой от интерфейса или порта клиента к импортируемому интерфейсу.

Когда диаграмма компонентов используется, чтобы показать внутреннюю структуру компонентов, предоставляемый и требуемый интерфейсы составного компонента, могут делегироваться в соответствующие интерфейсы внутренних компонентов.

Запрос пользователя поступает в компонент Application, где происходит обработка запроса с вызовом соответствующего контроллера и инициализация других компонентов. После этого компонент User загружает данные пользователя, которые используются во всей системе. Затем контроллером обрабатывается запрос и генерируется ответ. Для обращения к базе данных используется компонент ActiveRecord, который через компонент QueryBuilder посылает необходимые запросы к СУБД. После получения результатов от компонента ActiveRecord, контроллер вставляет эти результаты в представления (компонент View) и в конечном итоге генерирует ответ.

2.5 Диаграмма развертывания проектируемой информационной системы

Диаграмма развёртывания (англ. Deployment diagram) в Unified Modeling Language (в дальнейшем UML) моделирует физическое развертывание артефактов на узлах [16]. Например, чтобы описать web-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты (узлы) существуют (например, web-сервер, сервер базы данных, сервер приложения), какие программные компоненты (артефакты) работают на каждом узле (например, web-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

Диаграмма развертывания системы отображена на рисунке 10. На ней представлен web-сервер, соединенный с СУБД с доступом к нему через

активный концентратор как локальных пользователей, так и удаленных. Удаленные пользователи подключаются к web-серверу через интернет.

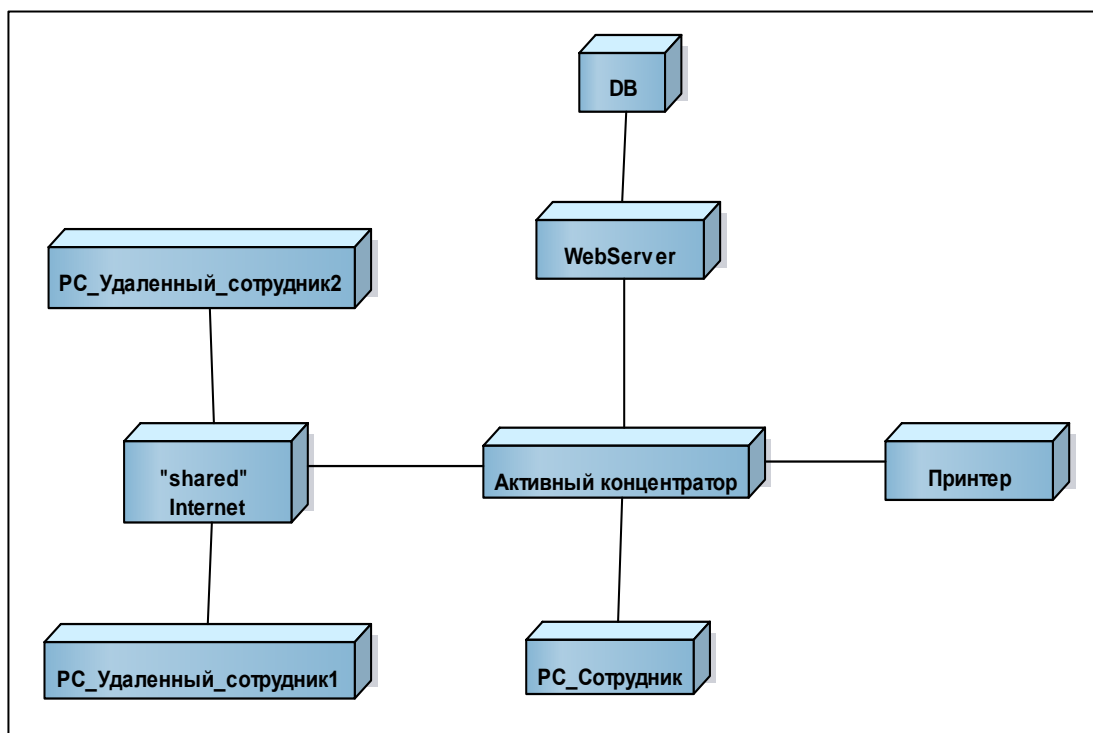


Рисунок 10 — Диаграмма развертывания проектируемой информационной системы

2.6 Диаграмма сущность-связь проектируемой информационной системы

Логическая модель данных описывает факты и объекты, подлежащие регистрации в будущей базе данных. Основными компонентами такой модели являются сущности, их атрибуты и связи между ними. Как правило, физическим аналогом сущности в будущей базе данных является таблица, а физическим аналогом атрибута — поле этой таблицы. С логической точки зрения сущность представляет собой совокупность однотипных объектов или фактов, называемых экземплярами этой сущности [17].

Сущности логической модели данных:

- users;
- indications;

- news;
- request.

Атрибуты сущности users:

- id: int(11), первичный ключ;
- username: varchar(50);
- password: varchar(100);
- fio: text;
- auth_key: int(11);
- role: smallint(10).

Атрибуты сущности indications:

- user_id: int(11), первичный ключ, внешний ключ к первичному ключу таблицы users;
- hot_counter: float;
- cold_counter: float;
- date: date.

Атрибуты сущности news:

- id_news: int(11);
- title: varchar(255);
- briefly: varchar(255);
- full_news: text;
- date: date.

Атрибуты сущности request:

- id: int(11), первичный ключ;
- theam: varchar(255);
- text: text;
- status: varchar(255);
- date: date;
- user: varchar(255).

Диаграмма сущность-связь представлена на рисунке 11.

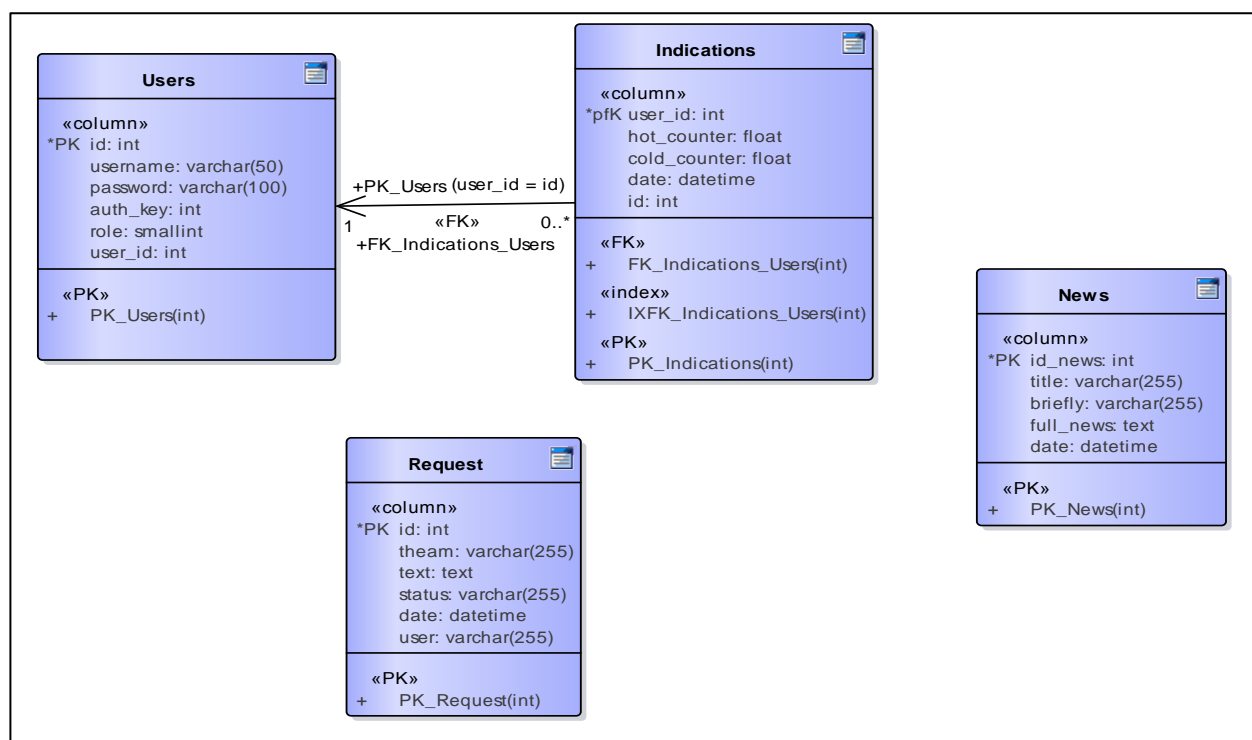


Рисунок 11 — Диаграмма сущность-связь проектируемой информационной системы

2.7 Диаграмма последовательности проектируемой информационной системы

Диаграмма последовательности (англ. sequence diagram) — диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов).

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (англ. lifeline), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Диаграмма последовательности процесса отправки показаний счетчиков представлена на рисунке 12.

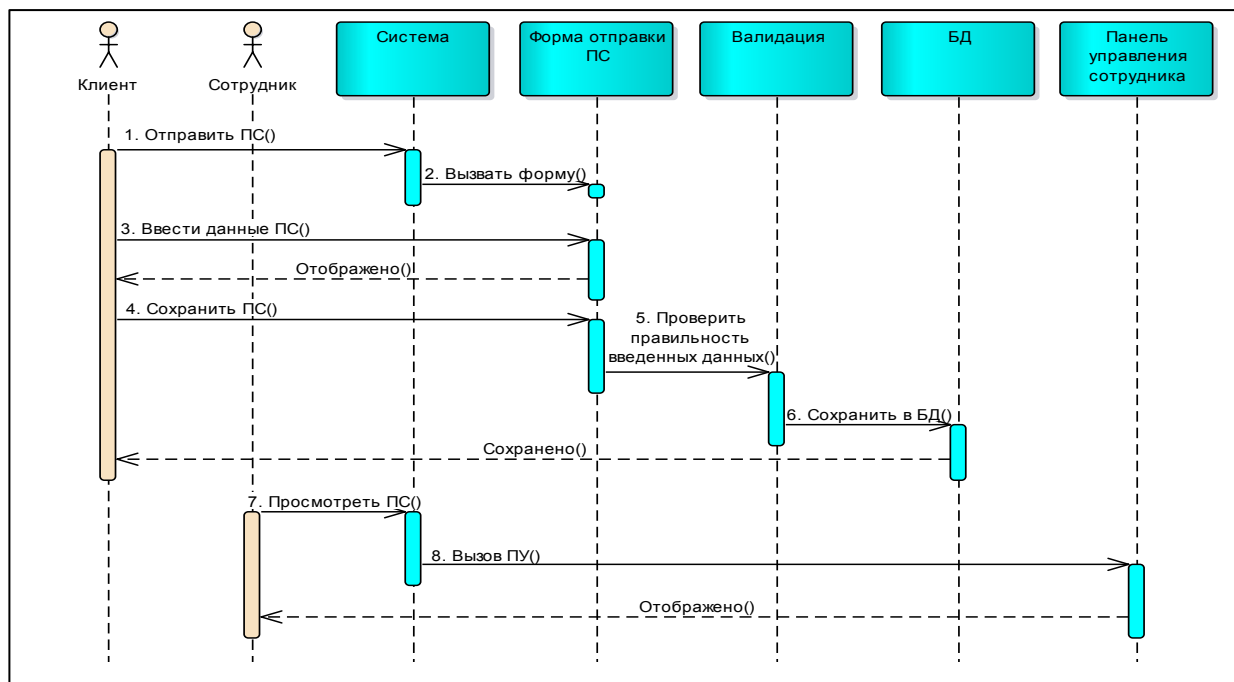


Рисунок 12 — Диаграмма последовательности процесса отправки показаний счетчиков

Диаграмма последовательности процесса отправки заявок на услуги представлена на рисунке 13.

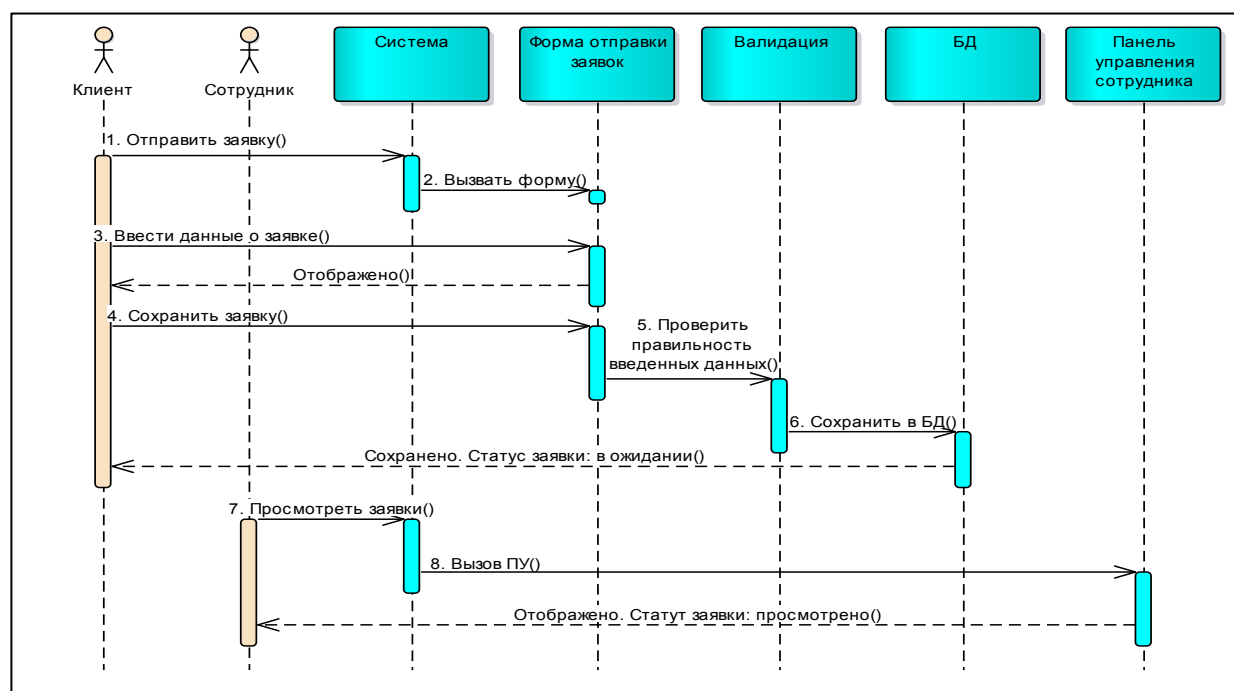


Рисунок 13 — Диаграмма последовательности процесса отправки заявок на услуги

2.8 Проектирование дизайна системы

Для проектирования дизайна системы были выбраны модульная сетка Bootstrap и программа Adobe Photoshop CC 2015.

Модульная система упрощает и ускоряет художественное конструирование и создаёт благоприятные условия для автоматизации вёрстки при использовании компьютерных настольно-издательских систем.

При верстке адаптивного классического макета: шапка сайта (header), основная часть (content), боковая колонка (sidebar) и нижняя часть сайта (footer), для корректного отображения необходимо рассчитать ширину в процентах каждого элемента. Для этих целей нужна сетка Bootstrap. Задаются классы для блоков, которые указывают, какую ширину должен занимать элемент и как он будет отображаться на разных устройствах. Сетка функционирует как таблица, в которой есть свои ряды и столбцы, максимальное количество столбцов 12. Сетку можно делать внутри другой сетки сколько угодно [19].

Макет сайта с использованием сетки Bootstrap представлен на рисунке 14.

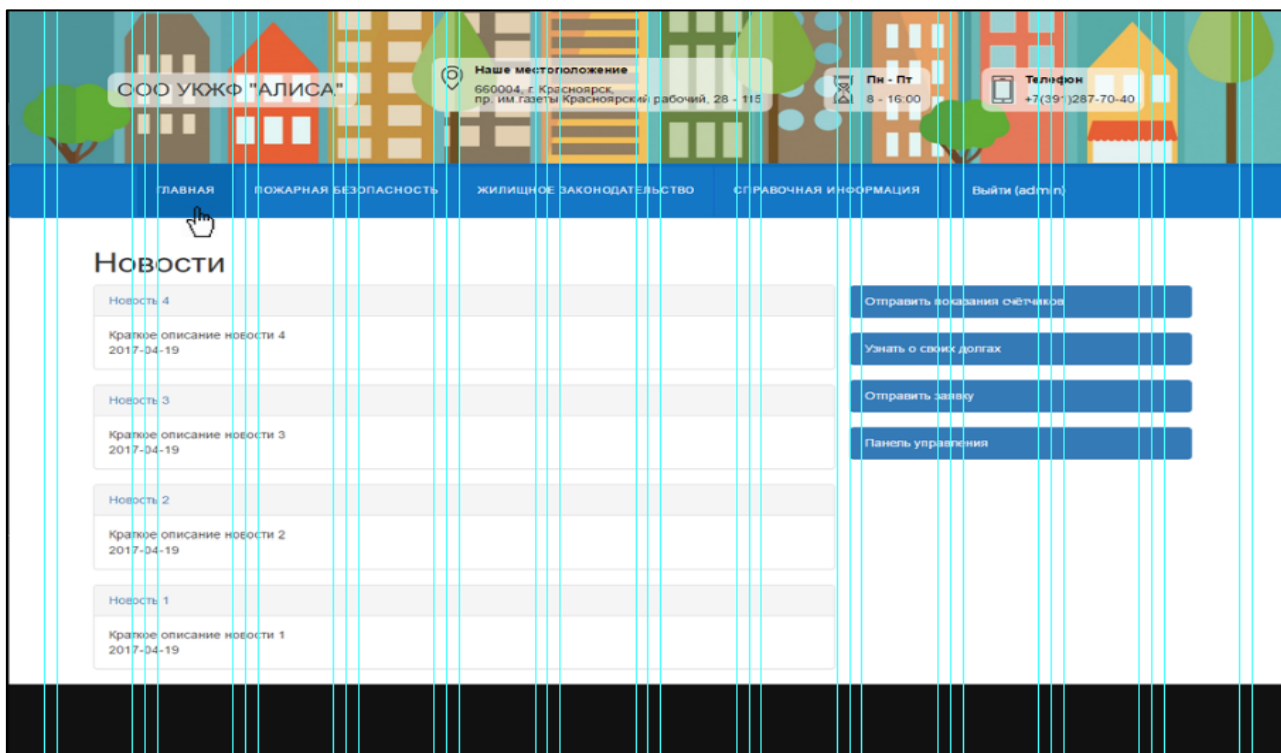


Рисунок 14 — Макет сайта

2.9 Выводы по главе 2

Выбор фреймворка Yii2 обусловлен эффективностью в использовании, простотой в освоении, а также широкими возможностями для разработки web-приложений. Гибкость и расширяемость фреймворка помогает вносить изменения в уже разработанный продукт в максимально короткие сроки, не останавливая при этом рабочий процесс. Также, для этого служит шаблон проектирования MVC, отделяя бизнес-логику от пользовательского интерфейса, позволяет изменять отдельные части приложения, не затрагивая другие.

Диаграмма вариантов использования показывает все возможные действия пользователей разрабатываемой ИС (администратор сайта, сотрудник компании, клиент). Диаграмма компонентов показывает разбиение программной системы на структурные компоненты и связи между компонентами. Диаграмма развертывания моделирует физическое развертывание артефактов на узлах. В представленной выше модели данных можно рассмотреть структуру базы данных информационной системы. База данных включает в себя 4 таблицы, исключая системные.

3 Разработка информационной системы с web-интерфейсом

После анализа выбранных средств проектирования, переходим к этапу разработки.

3.1 Настройка ролевого управления доступом

Список ролей:

- гость (guest) — не авторизованный пользователь;
- сотрудник (employee) — авторизованный пользователь сотрудник, имеет свои уникальные разрешения;
- клиент (client) — авторизованный пользователь клиент, имеет свои уникальные разрешения;
- администратор (administrator) — авторизованный пользователь администратор, имеет свои уникальные разрешения.

Дополнительные сведения о реализации:

- роль определяется таблицей users через значение role;
- роли имеют вложенную структуру — одна роль может наследовать разрешения другой;
- используется yii\rbac\PHPManager;
- генерирование конфигурации «роль-разрешения» необходимо выполнить с помощью предложенных разработчиком методов.

Предварительная настройка. Для функционирования ролевой системы доступа необходимо переопределить определенный компонент менеджера авторизации в конфигурационном файле @app/config/console.php для консольного приложения, которое будет генерировать файл ролей, и для конфигурационного файла @app/config/web.php, используемого web-приложением.

Как было описано выше, в ролевой модели доступа существуют операции, роли и задачи. Следующим шагом стало определения списка операций доступных на сайте:

- вход (login);
- выход (logout);
- ошибка (error);
- главная (index).

Действия с новостями:

- просмотр (view);
- создание (create);
- обновление (update);
- удаление (delete).

Действия с пользователями:

- просмотр (view);
- создание (create);
- обновление (update);
- удаление (delete).

Действия с жилищным законодательством и техникой безопасности:

- просмотр (view);
- загрузка (download).

Действия с заявками ну услуги:

- просмотр (view);
- создание (create);
- удаление (delete).

Действия с показаниями счетчиков:

- просмотр (view);
- создание (create);
- удаление(delete)
- обновление (update);
- загрузка (download).

Для каждой роли пользователя определены свои полномочия выполнения той или иной задачи.

Роли пользователей определены в базе данных в таблице users. Столбец role отвечает за распределение ролей. Строка данного столбца может принимать одно из следующих значений: клиент, сотрудник, администратор.

Класс `\app\models\Users` это компонент приложения. Описан как класс, отвечающий за идентификацию пользователя в конфигурационном файле.

Реализует интерфейс `\yii\web\IdentityInterface`, за счет чего позволяет идентифицировать пользователя, и наследует класс `\yii\db\ActiveRecord`, благодаря чему данные о пользователях могут содержаться в базе данных.

Класс содержит данные-члены и функции-члены. Каждая функция-член (метод) класса, как уже было сказано, имеет семантически обусловленное имя, которое сообщает о задаче, решаемой функцией.

Данные-члены класса в языке PHP называются свойствами.

Большинство методов класса используют так называемый неявный параметр `$this`. Этот параметр является указателем на объект класса, для которого был вызван данный метод. Далее дополнительно не указывается, что функция использует неявный аргумент, если это очевидно из остального ее описания.

Краткое описание. Класс `\app\models\Users` содержит свойства и предопределенные константы:

- `const ROLE_USER;`
- `const ROLE_COOPERATOR;`
- `const ROLE_ADMIN.`

Так как пароль в базе данных должен быть зашифрован, создано дополнительное свойство `password`, благодаря которому пользователь может ввести свой пароль для последующей шифрации.

Класс содержит методы:

- `public rules();`
- `public static function tableName();`

- public function attributeLabels ().

А так же обязательные свойства для реализации интерфейса `\yii\models\UserIdentity`:

- public static function findIdentity(\$id);
- public static function findIdentityByAccessToken(\$token, \$type = null);
- public function getId();
- public function getAuthKey();
- public function validateAuthKey(\$authKey);
- public function validatePassword(\$password).

Метод `rules()` в этом классе и ниже — метод, наследуемый от `\yii\db\ActiveRecord`, определяющий правила валидации данных перед записью в базу данных.

Метод `findIdentity ($id)` — метод, созданный с целью поиска пользователя в базе данных по его `id`, возвращает модель пользователя или `false` в случае неудачного поиска.

Метод `validatePassword ($password)` — метод, созданный с целью проверки соответствия пароля введенного пользователем с клавиатуры, с паролем хранящимся в зашифрованном виде в базе данных. Использует встроенную в Yii2 функцию сравнения.

Класс `app\models\LoginForm` наследует класс `yii\base\Model` и служит для авторизации пользователя в системе.

Класс `\app\models\LoginForm` содержит свойства:

- public \$username;
- public \$password;
- public \$rememberMe.

Свойства `$username` и `$password` содержат логин и пароль пользователя после ввода и отправки их на сервер. Свойство `$rememberMe` — булевый тип данных, содержит `true` или `false`, для определения, хочет ли пользователь оставаться в системе после закрытия браузера. Достигается это с помощью

установки долгосрочной cookie `_identity` содержащую зашифрованные данные пользователя.

Класс содержит методы:

- `public rules();`
- `public validatePassword($attribute, $params);`
- `public login();`
- `public getUser();`

Метод `getUser()` получает модель пользователя с базы данных, используя метод `\app\models\Users::findByUserName($username)` для поиска в БД.

Метод `validatePassword($attribute, $params)` проверяет пароль пользователя в базе данных с паролем введенным с клавиатуры, возвращает булевый тип. Для этого он использует методы `\app\models\LoginForm::getUser()`, который получает существующую модель пользователя, и метод `\app\models\Users::validatePassword($password)` для проверки пароля.

Метод `login()` авторизует пользователя в случае, если метод `\app\models\Users::validatePassword()` возвращает `true`. Является основным методом для запуска авторизации.

Класс `\app\controllers\SiteController` является базовым контроллером для остальных контроллеров в системе. Именно в нем проверяется возможность пользователя выполнять то или иное действие. При отказе в доступе пользователю выводится соответствующее сообщение.

3.2 Панель управления сотрудника

Класс `\app\controllers\AdminController` является контроллером для пользователей с правами сотрудник и администратор. В нем проверяется уровень доступа пользователя выполнять то или иное действие. При отказе в доступе пользователю выводится соответствующее сообщение.

Класс содержит методы:

- `public function actionIndex();`

- public function actionIndications();
- public function actionRequest();

Метод actionIndex() для вызова страницы рабочего места.

Метод actionIndications() для вывода страницы с данными по показаниям счетчиков.

Метод actionRequest() для вывода страницы с данными о заявках на услуги.

3.3 Интерфейсы

Для отображения списка меню в главном шаблоне системы используются виджеты Nav и NavBar из классов yii\bootstrap\Nav и yii\bootstrap\NavBar. Меню должно различаться для разных ролей пользователей. Чтобы не дублировать код и не использовать множество условий, отображение пунктов меню использует те же права доступа, которые прописаны в базе данных для данного пользователя.

Содержание меню заключается между begin() и end() с использованием виджета NavBar. Затем с помощью виджета Nav выводится список меню, используя уже прописанные атрибуты bootstrap:

- options – используется для создания контейнера для обертки меню;
- items – предназначен для создания списка меню;
- label – название пункта меню;
- url – адрес страницы на которую введет ссылка.

Используется контроллер SiteController, который содержит в себе методы:

- public function behaviors();
- public function actionIndex();
- public function actionView();
- public function actionLogin();
- public function actionLogout();
- public function actionContact();
- public function actionLegislation;
- public function actionSafety_measures;

- public function actionReference;
- public function actionIndications().

Метод public function behaviors() отвечает за передачу правил.

Метод public function actionIndex() предназначен для отображения главной страницы системы с новостями, отфильтрованными в обратном порядке.

Метод public function actionView() используется для вывода ошибки 404 при неверном пути.

Метод public function actionLogin() используется для авторизации пользователя в системе.

Метод public function actionLogout() используется для выхода пользователя из системы.

Метод public function actionContact() используется для отправки заявок на услуги.

Метод public function actionLegislation() используется для отображения страницы с информацией о жилищном законодательстве.

Метод public function actionSafety_measures() используется для отображения страницы с информацией о технике безопасности.

Метод public function actionReference() используется для отображения страницы со справочным материалом о компании.

Метод public function actionIndications() используется для отображения страницы с формой для отправки показаний счетчика.

При переходе по ссылке на страницу, пользователи с ролями «Клиент» и «Гость» попадают на главную страницу системы, где отображены последние новости компании.

В правой части страницы расположены функции, которыми может воспользоваться «Клиент» при успешной авторизации.

Главная страница системы представлена на рисунке 15.

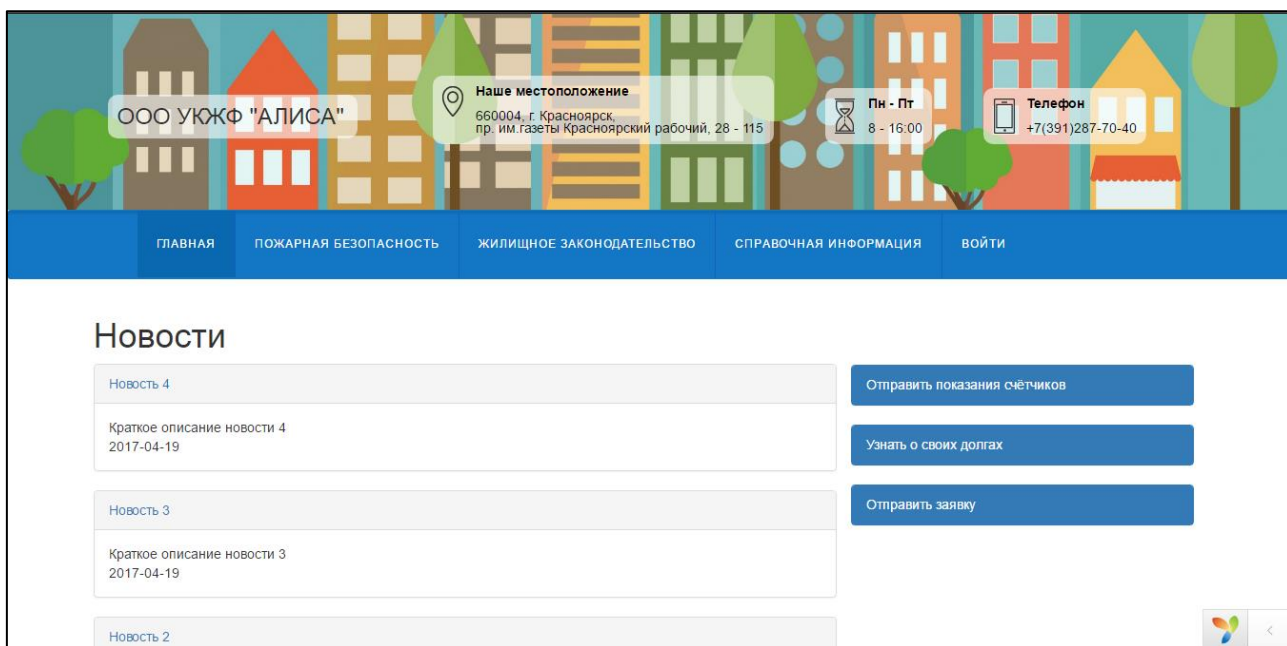


Рисунок 15 — Главная страница системы

В разделе «Отправить показания счетчиков» находится форма для отправки данных по показаниям счетчиков холодного и горячего водоснабжения. Представлено на рисунке 16.

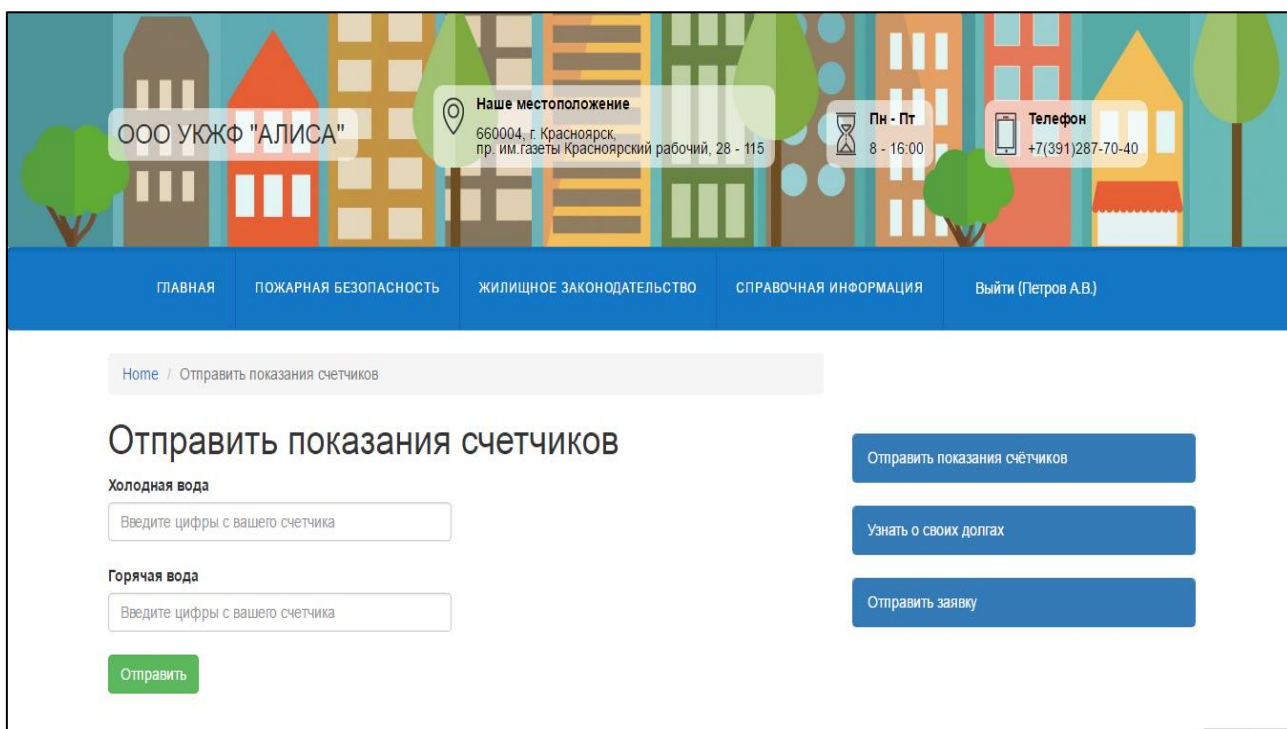


Рисунок 16 — Форма отправки показаний счетчиков

В разделе «Отправить заявку» находится форма для отправки заявок на услуги, необходимо указать тему заявки и, соответственно, текст заявки. Представлено на рисунке 17.

The screenshot shows a web interface with a blue header bar containing navigation links: 'ГЛАВНАЯ', 'ПОЖАРНАЯ БЕЗОПАСНОСТЬ', 'ЖИЛИЩНОЕ ЗАКОНОДАТЕЛЬСТВО', 'СПРАВОЧНАЯ ИНФОРМАЦИЯ', and 'Выйти (Петров А.В.)'. The main content area is white and contains a form for sending a request. The form has two input fields: 'Тема заявки' (Request Topic) and 'Подробности' (Details). Below the 'Подробности' field is a green 'Отправить' (Send) button. To the right of the form are three blue buttons: 'Отправить показания счётчиков' (Send meter readings), 'Узнать о своих долгах' (Learn about debts), and 'Отправить заявку' (Send request). Below the form, there is a preview of a submitted request titled 'Заявка 2'. The preview text reads: 'Прошу принять срочные меры по ремонту кровли в доме № 8 по улице Полевая в связи с тем, что она имеет множественные повреждения. Результатом разрушения кровельного покрытия стали многочисленные протечки в квартире 36, которая находится на верхнем этаже. 2017-05-16 Статус: В ожидании'.

Рисунок 17 — Форма отправки заявок на услуги

В системе разработана панель управления, которая появляется в меню у пользователя с ролью «Сотрудник» при успешной авторизации. С помощью панели управления «Сотрудник» имеет возможность обрабатывать показания счетчиков, заявки на услуги и добавлять новости. Панель управления отображена на рисунке 18.

The screenshot shows a web interface with a blue header bar containing navigation links: 'ГЛАВНАЯ', 'ПОЖАРНАЯ БЕЗОПАСНОСТЬ', 'ЖИЛИЩНОЕ ЗАКОНОДАТЕЛЬСТВО', 'СПРАВОЧНАЯ ИНФОРМАЦИЯ', and 'Выйти (Сергеев С.А.)'. The main content area is white and contains a management panel titled 'Панель управления'. The panel has three green buttons: 'Показания' (Readings), 'Заявки' (Requests), and 'Новости' (News). To the right of the panel are four blue buttons: 'Отправить показания счётчиков' (Send meter readings), 'Узнать о своих долгах' (Learn about debts), 'Отправить заявку' (Send request), and 'Панель управления' (Management panel). Above the header bar, there is a decorative banner with a cityscape illustration. The banner contains several information boxes: 'ООО УКЖФ "АЛИСА"', 'Наше местоположение' (Our location) with an address in Krasnoyarsk, 'Пн - Пт 8 - 16:00' (Monday - Friday 8 - 16:00), and 'Телефон +7(391)287-70-40' (Phone).

Рисунок 18 — Панель управления для «Сотрудник»

В разделе «Показания» выводится список, в котором отображены ФИО, номер лицевого счета, показания счетчиков и дата отправки показаний. Имеется возможность загрузки данной информации в память персонального компьютера в формате .xlsx. Представлено на рисунке 19.

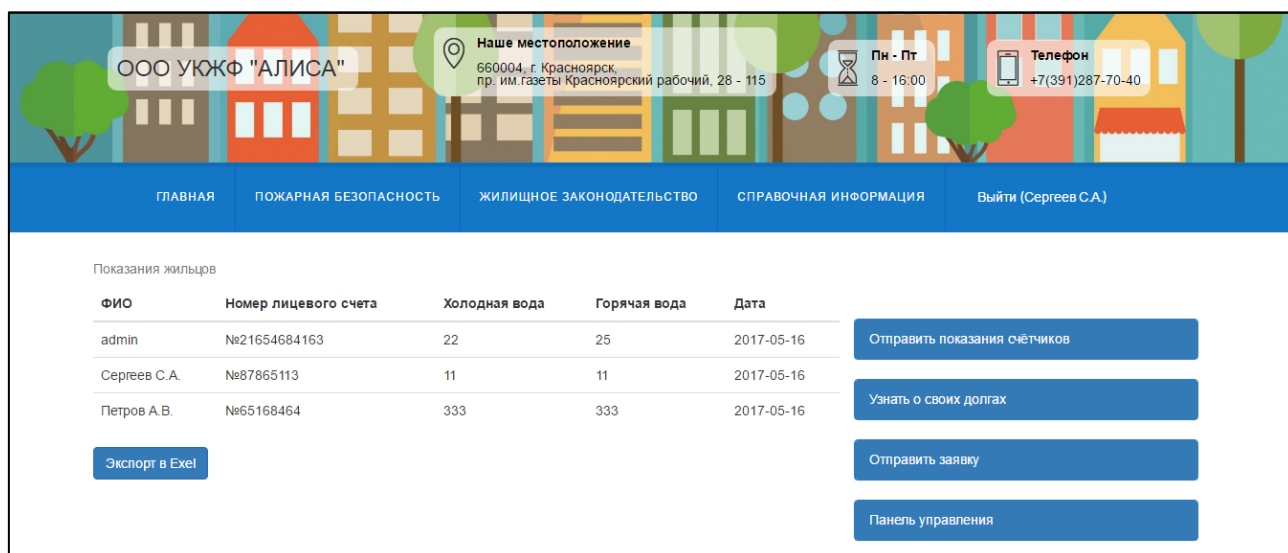


Рисунок 19 — Отчет по показаниям счетчиков

В разделе «Заявки» выводится список, в котором отображено от кого пришла заявка, текст заявки, дата и статус, как на рисунке 20.

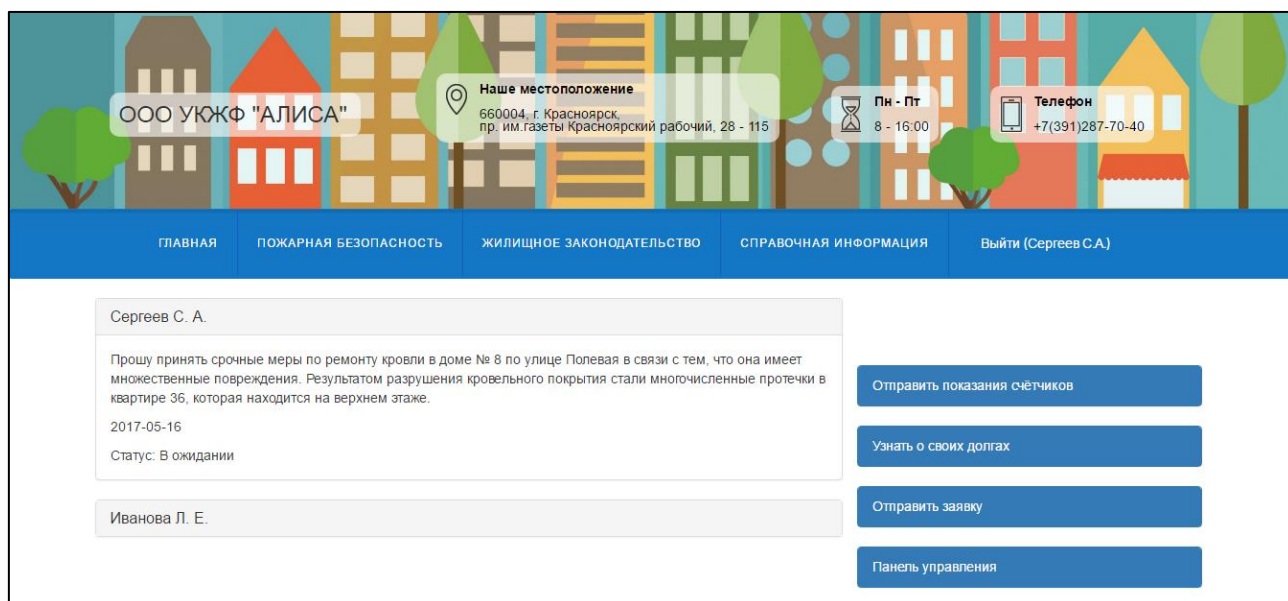


Рисунок 20 — Список заявок

Для роли «Администратор» доступен список пользователей в виде таблицы, в которой можно добавлять и удалять пользователей. Показано на рисунке 21.

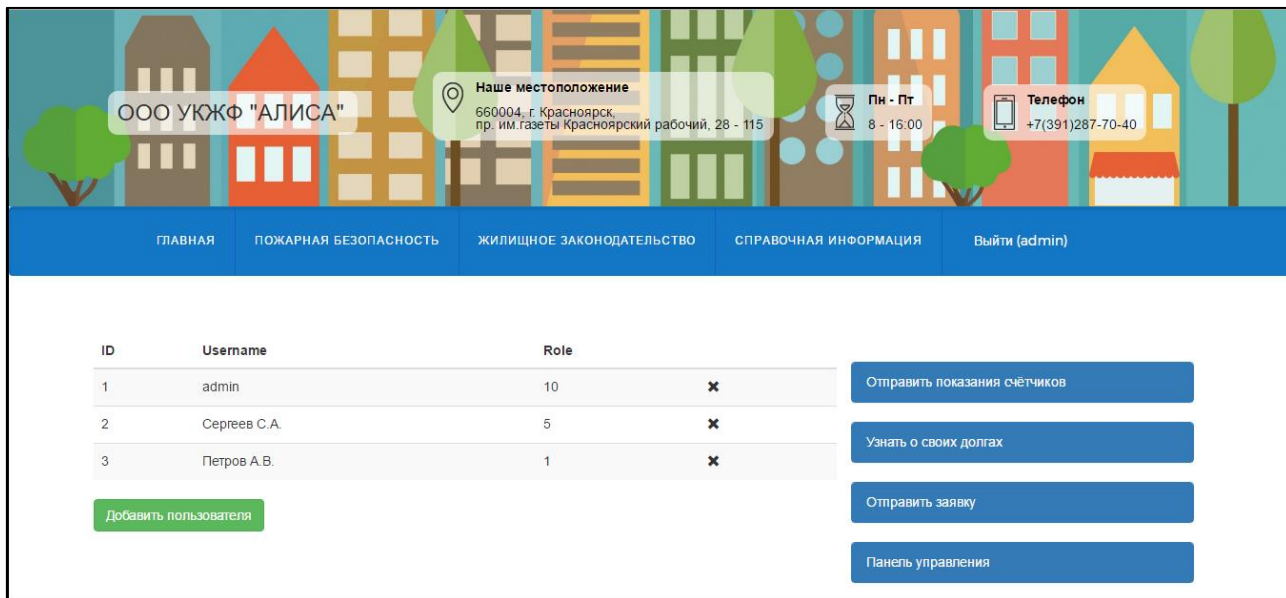


Рисунок 21 — Список пользователей

Также в системе хранится информация о жилищном законодательстве, тарифах, пожарной безопасности и справочная информация о компании.

Информация о жилищном законодательстве отображена списком на отдельной странице и доступна для загрузки. Показано на рисунке 22.

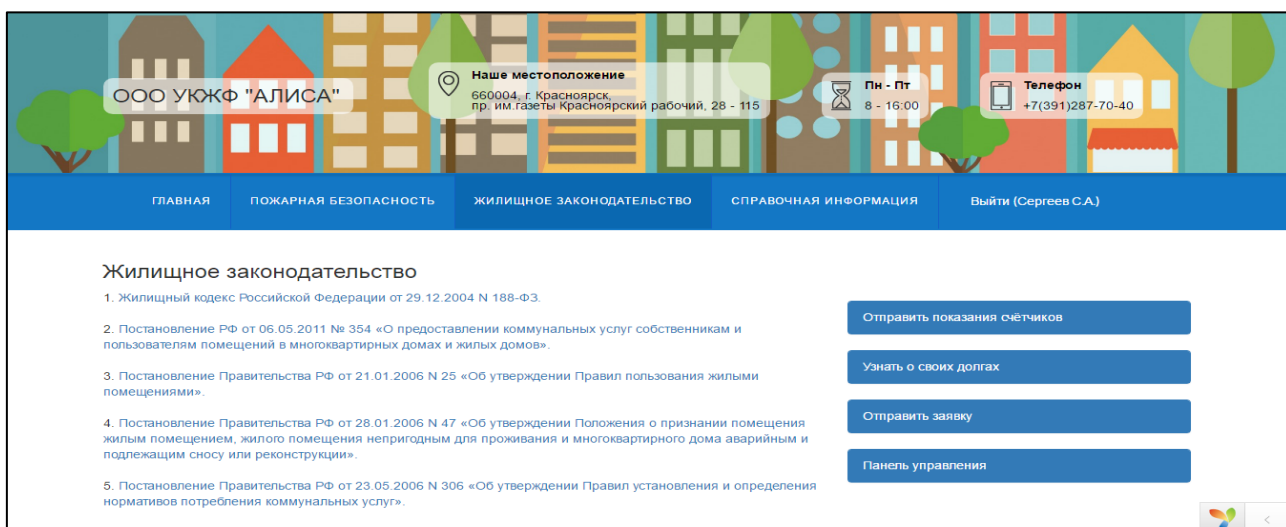


Рисунок 22 — Информация о жилищном законодательстве

Информация о пожарной безопасности также расположена отдельно, как на рисунке 23.

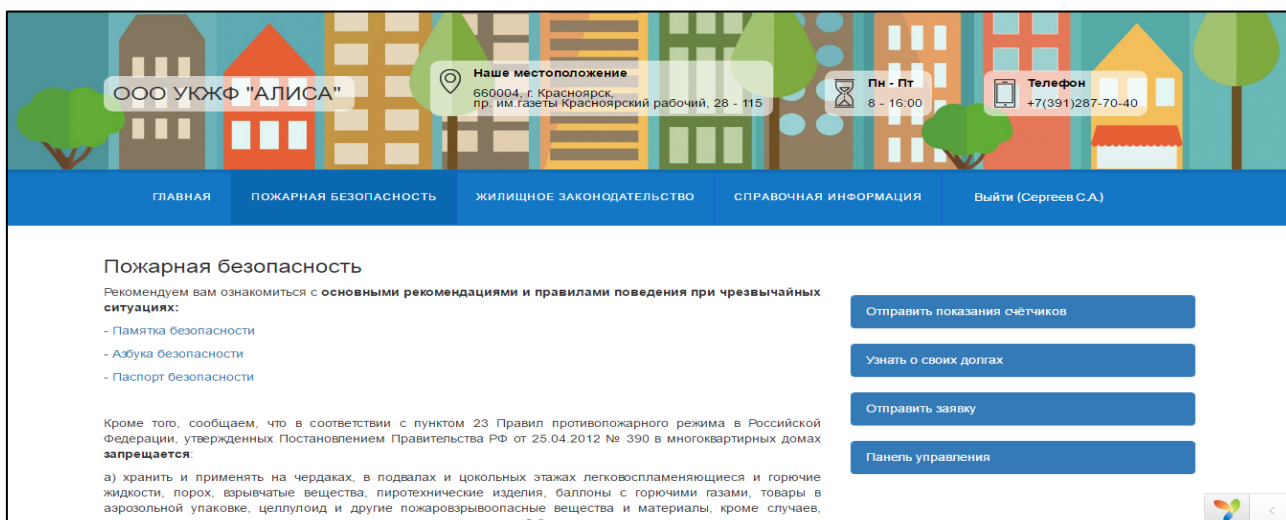


Рисунок 23 — Информация о пожарной безопасности

Справочная информация о компании отображена на рисунке 24.

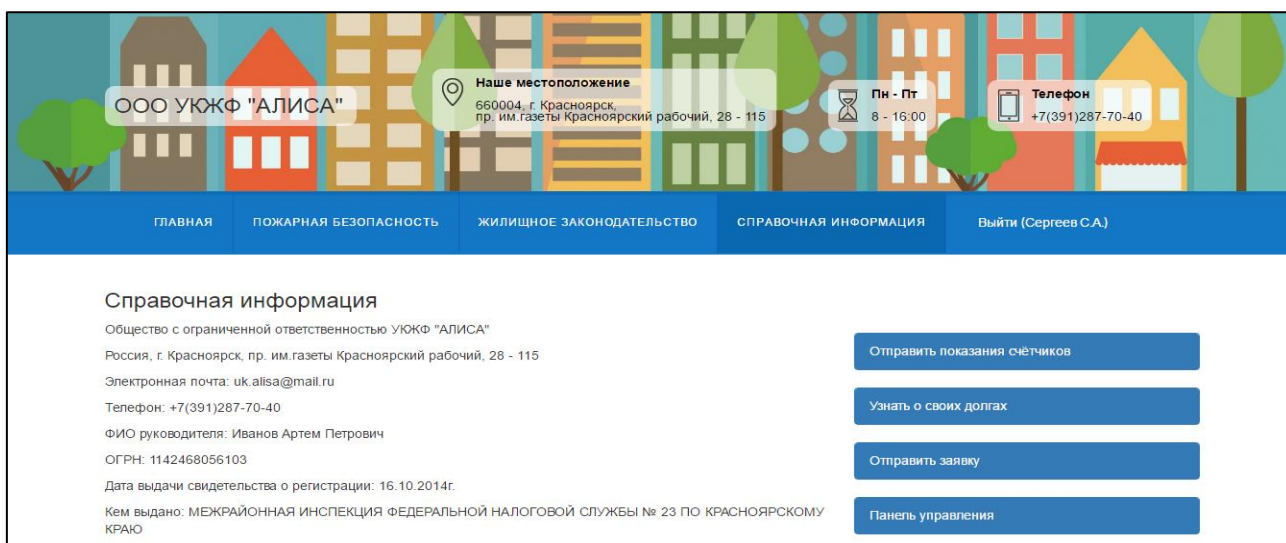


Рисунок 24 — Информация о компании

3.4 Выводы по главе 3

В ходе выполнения выпускной квалификационной работы была разработана информационная система с web-интерфейсом для управляющей компании ООО «Алиса», г. Красноярск, удовлетворяющая всем требованиям.

Рольное управление доступом является достаточно удобной организацией доступа и позволяет ограничить возможности каждого пользователя системы, что, в свою очередь, повышает информационную безопасность.

Панель управления сотрудника является важной функцией системы, она необходима для обработки данных по показаниям счетчиков и заявок на услуги.

Интерфейс системы отличается простотой, это необходимо для того, чтобы не нагружать рабочее пространство. Простота и удобство использования делают данную систему легко осваиваемой для специалистов, не имеющих образования в области информационных технологий.

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы были решены следующие задачи:

- а) анализ предметной области;
- б) проектирование информационной системы;
- в) разработка информационной системы с web-интерфейсом.

Для упрощения рабочего процесса, а также, увеличения производительности, необходимо создание информационной системы, отвечающей всем требованиям сотрудников и клиентов организации.

Предлагаемый проект информационной системы удаленного доступа соответствует поставленным задачам.

Система проектируется для управляющей компании ООО «Алиса» в г. Красноярске. Пользователями проектируемой ИС являются: клиент, сотрудник компании и администратор системы. Функциональные возможности ИС: функция аутентификации и авторизации, функции создания новой записи в БД, функции редактирования записей в БД путем ввода, замены, удаления содержимого записи с применением стандартных устройств ввода, в имеющиеся формы шаблонов, функции редактирования записи в БД с применением буфера обмена операционной системы, функции поиска информации по БД, функции формирования отчетности, функции отображения названия программы, версии программы, копирайта.

Главным плюсом проектируемой системы является возможность удаленного доступа, а также, размещение баз данных на сервере, что позволяет уменьшать нагрузку на устройства, которые используются локально.

В случае успешного внедрения проектируемой системы ожидается повышение производительности управляющей компании ООО «Алиса» в г. Красноярске за счет автоматизации процесса отправки показаний счетчиков, заявок на услуги, а также за счет своевременного оповещения клиентов о предстоящих событиях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Управляющая компания ЖКХ [Электронный ресурс]: – Электрон. текстовые дан. – Москва, 2009. – Режим доступа: http://ack1.ru/Library/upravlyaushchie_kompanii_ZHKN.html
2. Описание 1С: Учет в управляющих компаниях ЖКХ, ТСЖ и ЖСК [Электронный ресурс]: – Электрон. текстовые дан.– Москва, 2011. – Режим доступа: <http://solutions.1c.ru/catalog/jkh-tsjs>.
3. Инженерно-внедренческий Центр ИНСОФТ [Электронный ресурс]: – Электрон. текстовые дан. – Москва, 1999. – Режим доступа: http://insoft.ru/insoft/products/products_ais_JKH.htm.
4. ООО ГУК «Жилфонд» [Электронный ресурс]: – Электрон. Текстовые дан. – Красноярск, 2012. – Режим доступа: <http://www.krasguk.ru>.
5. Многофункциональный электронный портал ЖКХ Новосибирской области [Электронный ресурс]: – Электрон. текстовые дан. – Новосибирск, 2012. – Режим доступа: <https://жкхнсо.рф>.
6. Батоврин, В. К. Системная и программная инженерия: учебное пособие для вузов / В. К. Батоврин. – Москва: ДМК Пресс, 2010. - 280 с.
7. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы: Учебник / В. А. Гвоздева. – Москва: НИЦ ИНФРА-М, 2015. – 544 с.
8. Платонова В.Л., Отдельные аспекты защиты web-приложений // Сборник научных статей молодых ученых и студентов Тамб. Гос. Техн. Ун-та. – Тамбов, 2007. – Вып. 20 – С. 137.В
9. Васильков А.В., Безопасность и управление доступом в информационных системах: учебное пособие / А.В. Васильков, И.А. Васильков. – Москва: НИЦ ИНФРА-М, 2013. – 368 с.
10. Гришин В. Н., Информационные технологии в профессиональной деятельности: Учебник / В. Н. Гришин, Е. Е. Панфилова. – Москва: ИНФРА-М, 2015. – 416 с.

11. Фреймворк // Википедия. [2017—2017]. Дата обновления: 25.04.2017. URL: <http://ru.wikipedia.org/?oldid=85052657>.
12. Емельянова, Е. З. Проектирование информационных систем: учебное пособие / Е. З. Емельянова, Т. Л. Партыка, И. И. Попов. – Москва: НИЦ ИНФРА-М, 2014. – 432 с.
13. Заботина, Н. Н. Проектирование информационных систем: учебное пособие / Н. Н. Заботина. – Москва: ИНФРА-М, 2011. – 331 с.
14. Гринберг, А. С. Информационный менеджмент: учебное пособие для вузов / А. С. Гринберг, И. А. Король. – Москва: ЮНИТИ-ДАНА, 2012. – 415 с.
15. Прецедент (UML) // Википедия. [2016—2016]. Дата обновления: 23.02.2016. URL: <http://ru.wikipedia.org/?oldid=76699184>.
16. Киреева, Г. И. Основы информационных технологий: учебное пособие / Г. И. Киреева, В. Д. Курушин, А. Б. Мосягин. – Москва: ДМК Пресс, 2010. – 272 с.
17. Гайдамакин Н. А. Автоматизированные информационные системы, базы и банки данных. Вводный курс: Учебное пособие / Н. А. Гайдамакин. – Москва: Гелиос АРВ, 2002. – 368 с.
18. Штатное расписание управляющей компании [Электронный ресурс] : – Электрон. текстовые дан. – Москва, 2010. – Режим доступа: <http://101urist.com/nedvizhimost>.
19. Модульная система вёрстки // Википедия. [2017—2017]. Дата обновления: 06.03.2017. URL: <http://ru.wikipedia.org/?oldid=84094857>.

ПРИЛОЖЕНИЕ А

Техническое задание

АННОТАЦИЯ

В настоящем документе определены требования и порядок выполнения научно-исследовательских и опытно-конструкторских работ по созданию информационной системы с web-интерфейсом для управляющей компании ООО «Алиса», г. Красноярск, в соответствии с которыми будет проводиться разработка информационной системы (далее — ИС) и ее приемка при вводе в эксплуатацию.

1 Общие сведения

1.1 Полное наименование Системы и её условное обозначение

Полное наименование Системы: Информационная система с web-интерфейсом для управляющей компании ООО «Алиса» г. Красноярск.

Условное обозначение Системы: «ИС Алиса».

Далее по тексту также используется сокращенное условное обозначение «ИС Алиса» и «Система».

2 Назначение и цели создания Системы

2.1 Назначение Системы

Создаваемая Система предназначена для автоматизации процессов: рассылки необходимой информации клиентам, хранения и обработки данных о показаниях счётчиков клиентов, а также поступающих от них заявок.

2.2 Цели создания Системы

Система должна обеспечить достижение следующих целей:

- повышение эффективности рабочего процесса;
- повышение скорости информационного обмена внутри компании;
- повышение оперативности доступа к информации вне организации.

3 Характеристика объекта автоматизации

3.1 Краткие сведения об объекте автоматизации

ООО «Алиса» является обществом с ограниченной ответственностью, оказывающим жилищно-коммунальные услуги. «Алиса» действует на рынке города Красноярск. Программный продукт разрабатывается для сотрудников головного офиса, а также для клиентов, обслуживаемых данной компанией.

В таблице А.1 представлены ориентировочные данные о количественном составе работников организации [18].

Таблица А.1 — Штат организации

Профессия работника	Количество, (чел.)
Директор	1
Секретарь	1
Главный экономист	1
Главный бухгалтер	1
Бухгалтер по квартплате	2
Бухгалтер-кассир	3
Главный инженер	1
Инженер ПТО и ОТ	1
Инженер-энергетик	2
Инженер-ремонтник	5
Юрисконсульт	1
Начальник ЖЭУ	1

Особенностями бизнес-процессов и хозяйственных задач организации является то, что:

- задачи часто могут быть комплексными, и в их решении могут быть задействованы несколько сотрудников;

- для многих задач качество и оперативность их получения существенно влияют на эффективность их решения.

Результаты анализа существующего информационного обмена показали, что на настоящий момент, информационное взаимодействие между сотрудниками управляющей компании и клиентами происходит по телефонной связи, что тормозит рабочий процесс.

Специфическими особенностями процедур сбора, хранения и обработки указанных данных в настоящий момент являются:

- дублирование и противоречивость данных вследствие подготовки данных разными исполнителями на бумажных носителях;

- наличие значительного объема данных на бумажных носителях.

4 Требования к Системе

В данном разделе приведены все необходимые требования к разрабатываемой информационной системе.

4.1 Перспективы развития. Модернизация Системы

Проектные решения, применяемые при разработке Системы, должны обеспечивать возможность дальнейшего развития Системы. При развитии Системы принятая архитектура системы не должна изменяться.

Должна быть предусмотрена возможность дальнейшего развития Системы в следующих направлениях:

- расширение прикладных функций;

- интеграция Системы с другими информационными системами и ресурсами;

- увеличение ресурсов программно-аппаратного комплекса в соответствии с ростом потоков информации в Системе.

В процессе развития Системы предполагается возможность увеличения доли информации, которая хранится в Системе в структурированном виде.

Расширение функциональности Системы возможно в процессе разработки сервисов, обеспечивающих экспорт данных для системы формирования аналитической отчётности, а также других сервисов, обусловленных необходимостью обеспечения информационного обмена с другими системами.

4.2 Требования к численности и квалификации персонала Системы и режиму его работы

Персонал Системы обеспечивает выполнение административных действий в Системе, обновление и создание информации.

Детальная спецификация системы прав доступа для пользовательских ролей должна быть выполнена на стадии разработки технического проекта Системы. В число ролей пользователей Системы должны входить, как минимум, следующие роли:

- администратор Системы;
- сотрудник компании;
- клиент.

Состав персонала, работающего с Системой через её пользовательский интерфейс, представлен в таблице А.2.

Таблица А.2 — Подсистемы и их пользователи

Наименование подсистемы	Пользователи системы
Подсистема информационной безопасности	Администратор
Администрирование	Администратор
Хранение информации	Сотрудник, Администратор, Клиент
Информационное обеспечение клиентов	Сотрудник

При внедрении Системы состав пользователей подсистем, их роли и режим работы должны быть уточнены для каждого подразделения, эксплуатирующего Систему.

В таблице А.3 представлены уровни классификации, которым должны соответствовать пользователи Системы.

Таблица А.3 — Уровни квалификации пользователей

Уровень квалификации	Требования
Пользователь низкой квалификации	Умение включать персональный компьютер на базе операционных систем: Windows 7 и AltLinux 6.0, а также проходить процесс аутентификации
Пользователь средней квалификации	Опыт работы с персональным компьютером на базе операционных систем: Windows 7 и AltLinux 6.0
Квалифицированный специалист	Высшее техническое образование, стаж работы не менее 2 лет, знание английского языка (технический перевод), знание администрирования промышленной СУБД

Эксплуатация Системы должна проводиться с обученным персоналом. Обучение персонала должно быть проведено до начала ввода ее в эксплуатацию в соответствии с программой обучения.

Для поддержки функционирования Системы может быть организована ее эксплуатация с использованием услуг, как собственных специалистов, так и услуг по сопровождению с привлечением сторонних специалистов, обладающих знаниями в области информационных и сетевых платформ, на которых реализована Система, а также опытом администрирования крупных баз данных и операционных систем.

4.4 Требования к защите информации от несанкционированного доступа

Система относится к группе многопользовательских автоматизированных систем с разными правами доступа.

Подсистема информационной безопасности призвана реализовать всю базовую функциональность по защите информации, включая:

- обнаружение и предотвращение вторжений;
- контроль защищенности;
- разграничение прав доступа.

На стадии технического проектирования Системы должен быть разработан документ, определяющий политику информационной безопасности Системы. Политика информационной безопасности должна обеспечить регламентацию процессов по управлению информационной безопасностью при эксплуатации Системы. Документ «Политика обеспечения информационной безопасности» должен войти в состав эксплуатационной документации Системы.

В Системе должно быть обеспечено выполнение следующих требований:

Для пользователей из числа административного персонала Системы:

- доступ в систему через пользовательский интерфейс должен предоставляться только пользователям, успешно прошедшим процедуру аутентификации;

- должна выполняться журнализация входа/выхода пользователей, включая администраторов, в Систему с использованием пользовательского интерфейса Системы.

В параметрах записи журнала должны фиксироваться:

- дата и время входа/выхода пользователя в/из систему/ы;
- результат попытки входа (успешная или неуспешная);
- идентификатор пользователя;
- адрес исходящего запроса.

4.5 Требования к функциям, выполняемым Системой

Функции, которые должны быть реализованы в Системе (для сотрудника):

- авторизация;
- просмотр показаний счетчиков;

- экспорт отчета по показаниям счетчиков в формате .excel;
- просмотр заявок на услуги от клиентов;
- удаление заявок на услуги;
- добавление новостей на главной странице системы;
- просмотр новостей;
- удаление новостей.

Функции, которые должны быть реализованы в Системе (для клиентов):

- авторизация;
- отправка показаний счетчиков;
- обновление данных по показаниям счетчиков;
- просмотр данных по показаниям счетчиков;
- отправка заявок на услуги;
- просмотр заявок на услуги;
- просмотр информации в системе.

Функции, которые должны быть реализованы в Системе (для администратора):

- авторизация;
- просмотр пользователей ИС;
- создание пользователей ИС;
- обновление пользователей ИС;
- удаление пользователей ИС;
- просмотр ролей пользователей;
- создание ролей пользователей;
- обновление ролей пользователей;
- удаление ролей пользователей.

Функции, которые должны быть реализованы в Системе (для информационной безопасности вышеперечисленных пользователей):

- защита персональных данных пользователей;
- разграничение доступа к различным функциям.

4.6 Требования к лингвистическому обеспечению Системы

Моделирование предметной области, требований и архитектуры Системы, ее компонентов и их взаимодействия должно быть произведено с использованием стандартов в области разработки программного обеспечения.

Для текстовых полей Система должна иметь возможность поддерживать использование кодировки UTF-8. В системных диалогах с пользователями должен использоваться русский язык, исключением могут быть сообщения общесистемного программного обеспечения, так как они не подлежат переводу.

Система должна быть реализована с помощью языков программирования высокого уровня, имеющих промышленные масштабы развития и сопровождения.

В качестве языка манипулирования данными и языка определения данных в СУБД должен быть использован язык SQL.

4.7 Прикладное программное обеспечение

Прикладное программное обеспечение Системы должно обладать следующими свойствами:

- функциональная достаточность (полнота);
- модульность построения и удобство эксплуатации;
- использование двухуровневой архитектуры (клиент-сервер) распределенной системы.

В прикладном программном обеспечении должны быть реализованы меры по защите от ошибок при вводе и обработке данных.

Система должна представлять собой платформу-независимое приложение, включающее следующие уровни:

- уровень пользовательского интерфейса (UI) для обеспечения доступа пользователей к функциям Системы;

- уровень бизнес-логики (BusinessLogicLayer), представляющий собой набор программных сервисов, исполняемых на сервере приложений и реализующих всю бизнес-логику и общесистемные механизмы;

- уровень данных (DataLayer), который должен быть реализован на основе промышленной СУБД и сети хранения данных.

Разделение Системы на отдельные независимые уровни должно обеспечить возможность разворачивать Систему в самых разных конфигурациях (различные серверы приложений, СУБД, интеграционные решения).

4.8 Системное программное обеспечение

Требования к системному программному обеспечению приведены в таблице А.4.

Таблица А.4 — Требования к системному программному обеспечению

№	Категория ПО	Требования
1.	Интерфейс	Реализация пользовательского интерфейса должна обеспечить возможность использования стандартных браузеров: MozillaFirefox, GoogleChrome, Microsoft Internet Explorer.
2.	Сервер	Сервер должен представлять собой среду выполнения и инструменты управления приложениями, созданными на основе web-сервисов. Также должен обеспечивать совместимость с выбранной для использования СУБД.
3.	Система управления БД	СУБД должна обеспечивать: <ul style="list-style-type: none">- поддержка реляционной модели данных;- целостность данных;- возможность восстановления данных при сбоях;- возможность сжатия данных;- возможность разбиения индексов и таблицы на более простые компоненты;- возможность оптимизации SQL-запросов.
4.	HTTP-сервер	HTTP-сервер должен обеспечивать защищенное соединение с клиентским ПО, возможность кэширования контента и документов.

ПРИЛОЖЕНИЕ Б

Плакаты презентации



Рисунок Б.1 — Слайд презентации № 1

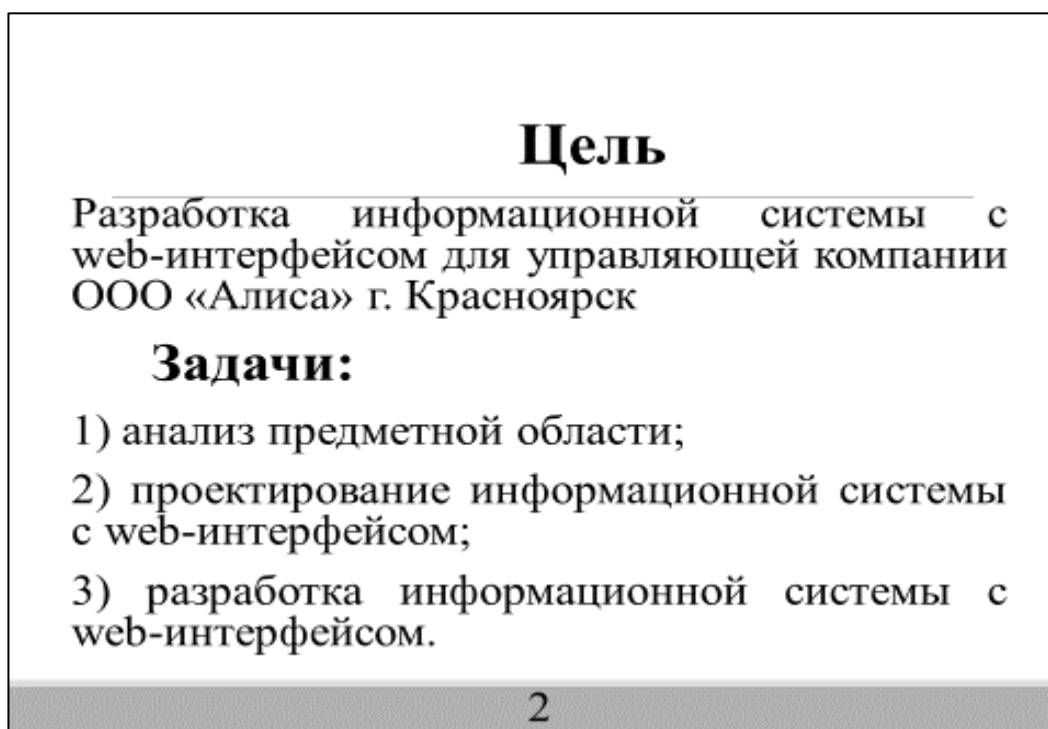


Рисунок Б.2 — Слайд презентации № 2

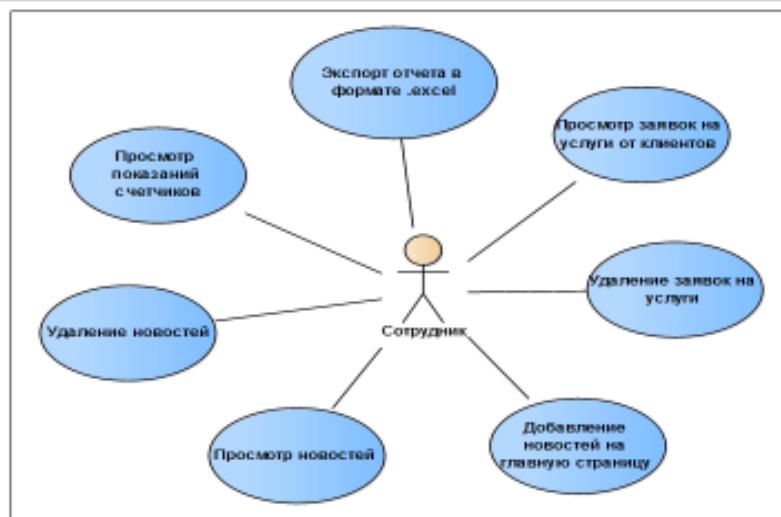
Актуальность

- своевременное информирование клиентов о новых событиях;
- возможность удаленного доступа к системе;
- возможность отправки показаний счетчиков и заявок на услуги.

3

Рисунок Б.3 — Слайд презентации № 3

Диаграмма вариантов использования для актора «Сотрудник»



4

Рисунок Б.4 — Слайд презентации № 4

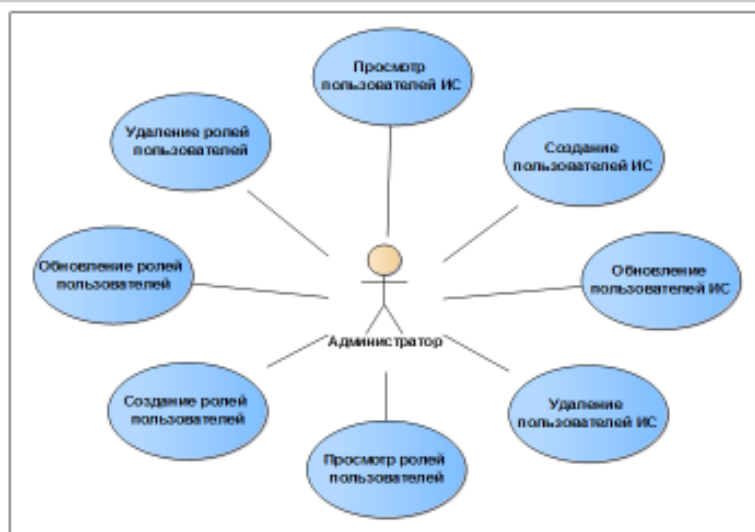
Диаграмма вариантов использования для актора «Клиент»



5

Рисунок Б.5 — Слайд презентации № 5

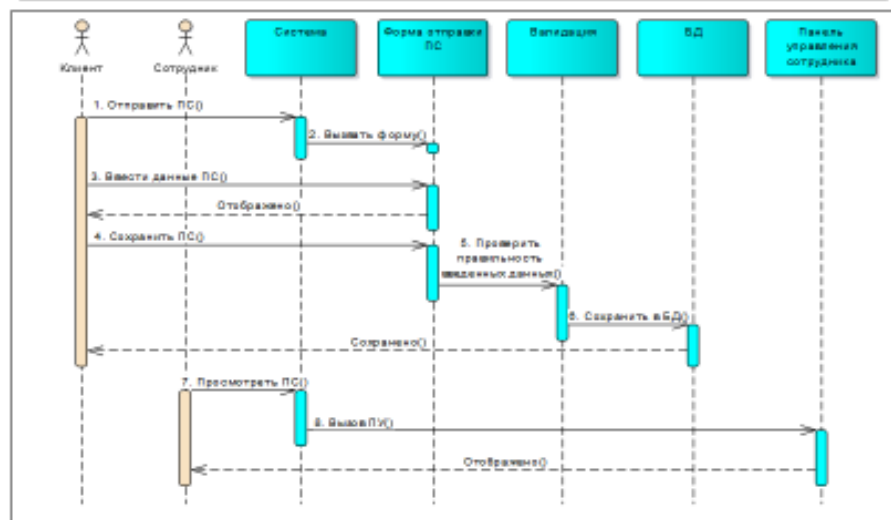
Диаграмма вариантов использования для актора «Администратор»



6

Рисунок Б.6 — Слайд презентации № 6

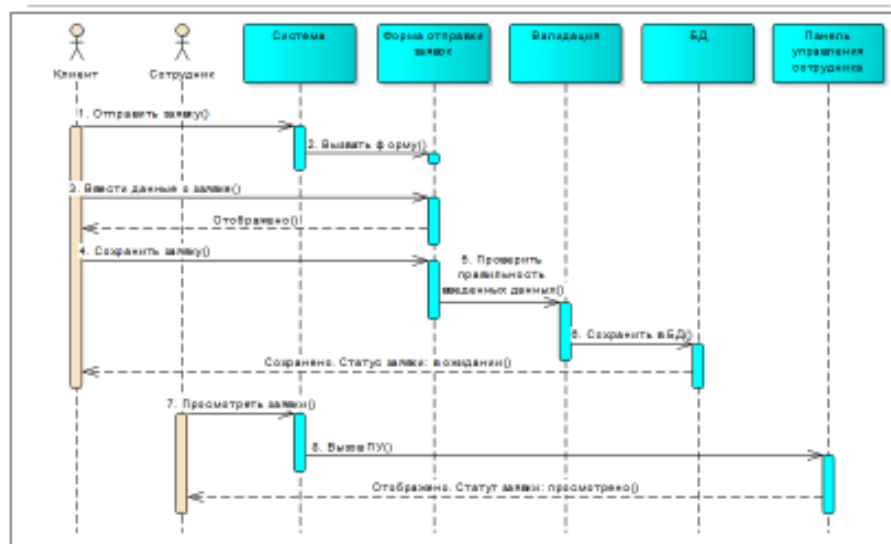
Диаграмма последовательности



7

Рисунок Б.7 — Слайд презентации № 7

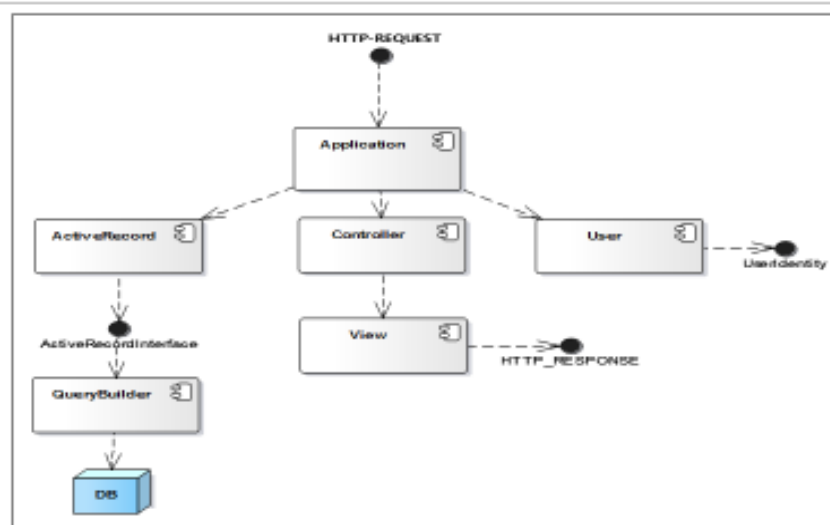
Диаграмма последовательности



8

Рисунок Б.8 — Слайд презентации № 8

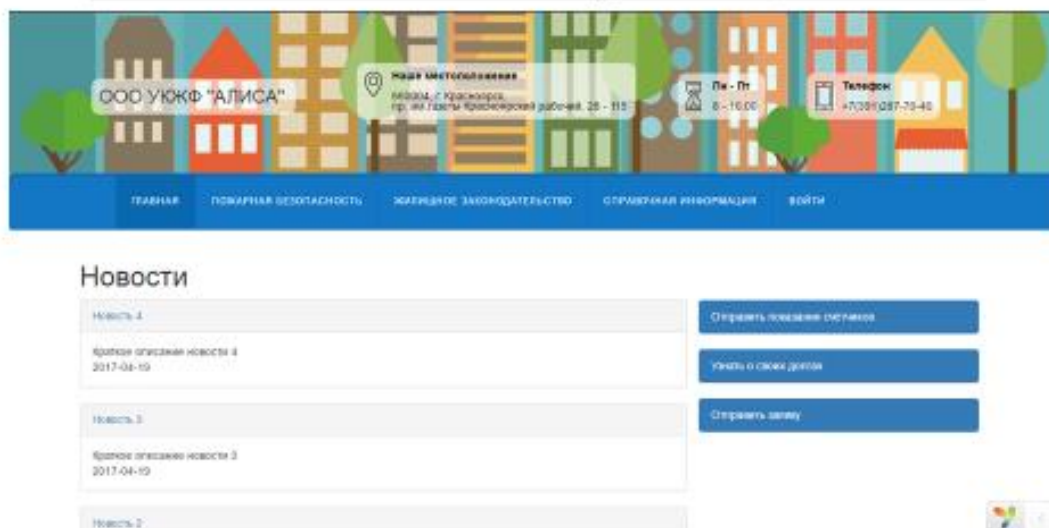
Диаграмма компонентов



9

Рисунок Б.9 — Слайд презентации № 9

Главная страница



10

Рисунок Б.10 — Слайд презентации № 10

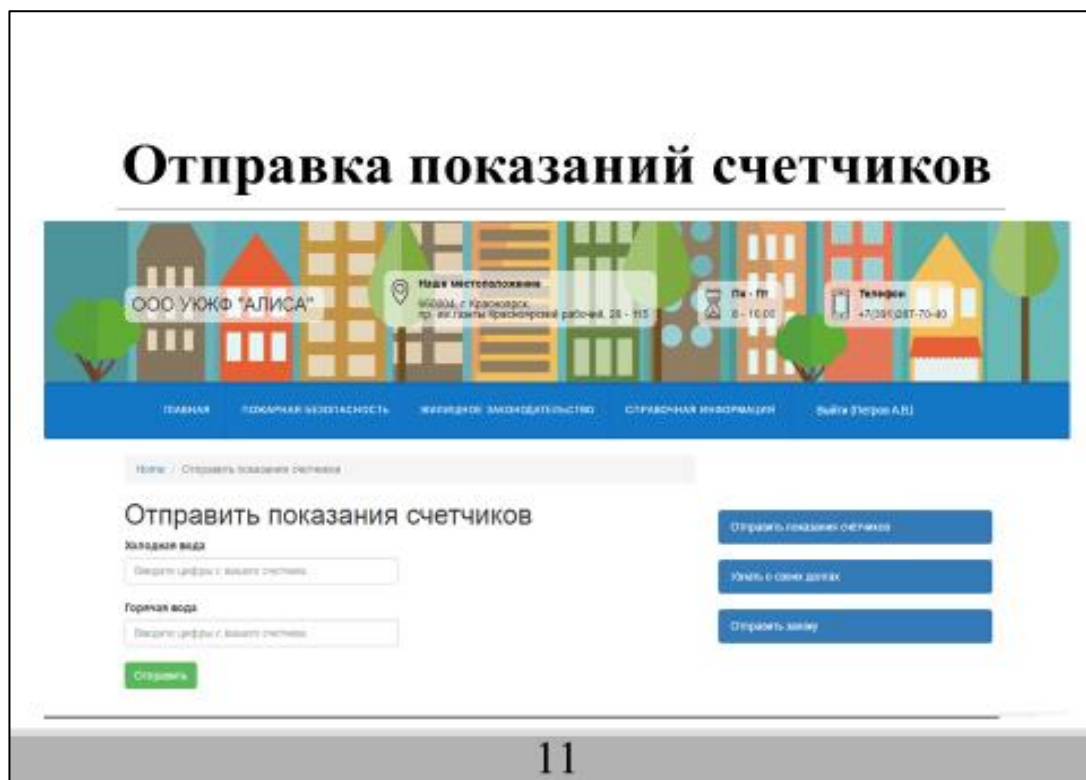


Рисунок Б.11 — Слайд презентации № 11

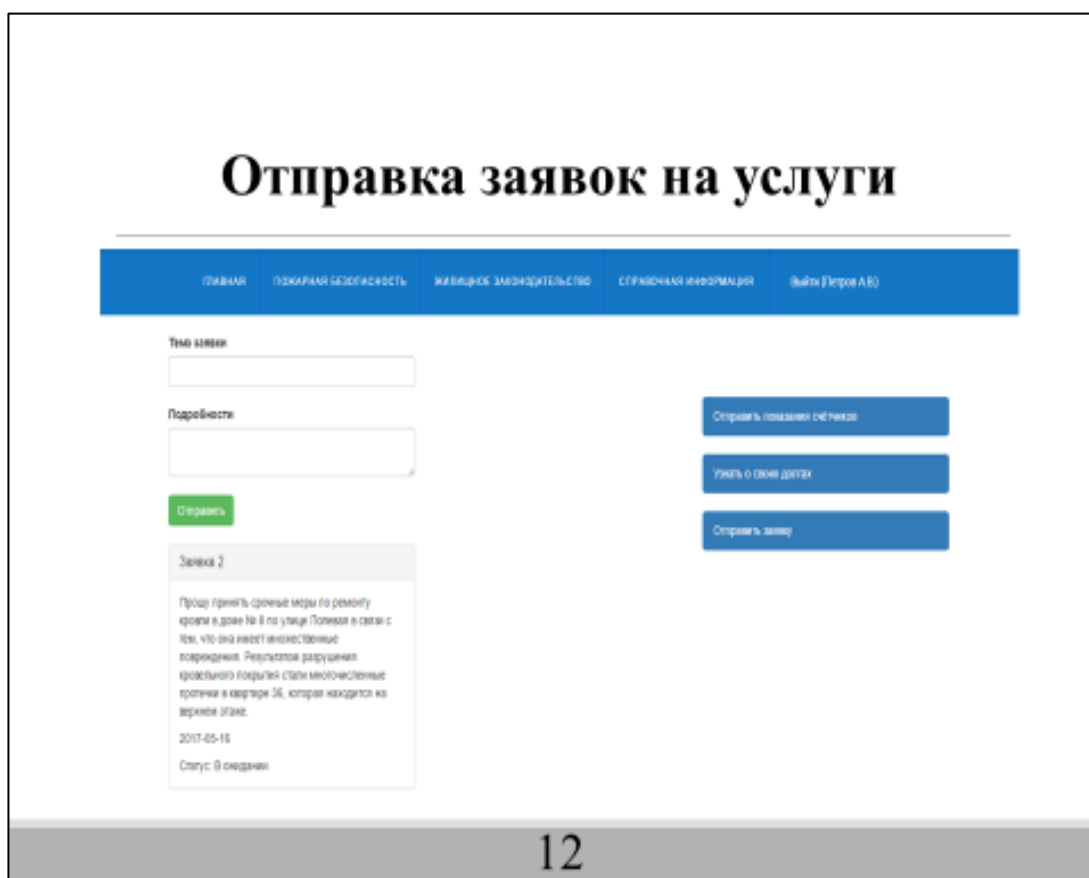
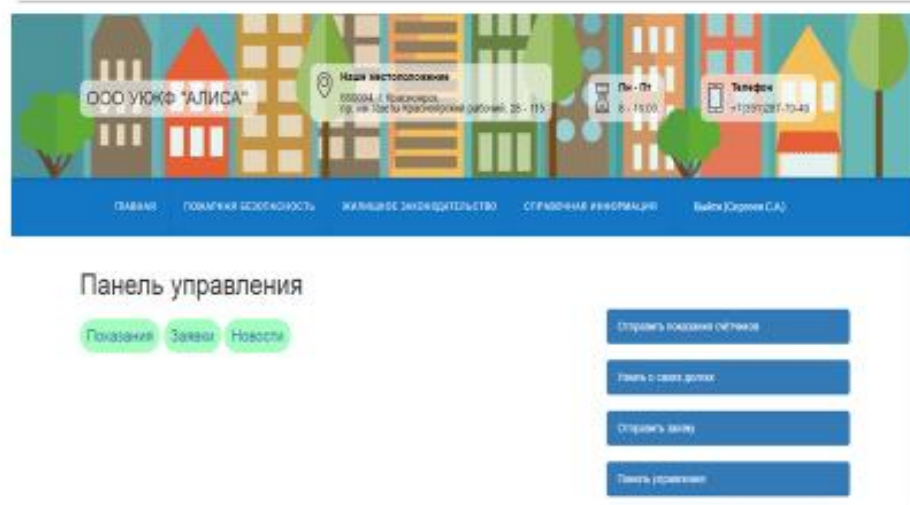


Рисунок Б.12 — Слайд презентации № 12

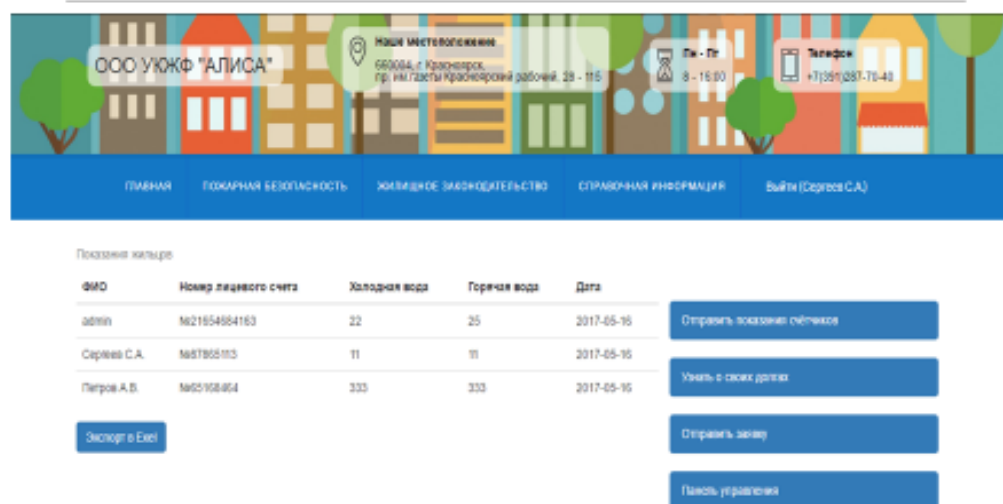
Панель управления администратора



13

Рисунок Б.13 — Слайд презентации № 13

Отчет по показаниям счетчиков



14

Рисунок Б.14 — Слайд презентации № 14

Список заявок на услуги

ООО УЮФФ "АЛИСА"

Место инцидента
000004, г. Красноярск, пр. им. Гайдук Красноярский район, 28 - 115

Пн - Пт
8 - 18:00

Телефон
+7(391)287-70-48

Главная Пожарная безопасность Жилищное законодательство Отраслевая информация Вышли (Сервис С.А.)

Сервис С. А.

Прошу принять срочные меры по ремонту кровли в доме № 8 по улице Полевая в связи с тем, что она имеет многочисленные повреждения. Результатом разращения кровельного покрытия стали многочисленные протечи в подвале 35, которая находится на первом этаже.

2817-65-16

Статус: В ожидании

Иванова Л. Е.

Отправить показания счётчиков

Уведомить о своем доходе

Отправить заявку

Получить уведомление

15

Рисунок Б.15 — Слайд презентации № 15

Список пользователей системы

ООО УЮФФ "АЛИСА"

Место инцидента
000004, г. Красноярск, пр. им. Гайдук Красноярский район, 28 - 115

Пн - Пт
8 - 18:00

Телефон
+7(391)287-70-48

Главная Пожарная безопасность Жилищное законодательство Отраслевая информация Вышли (admin)

ID	username	Role	
1	admin	10	✖
2	Сервис С.А.	5	✖
3	Петров А.В.	1	✖

Добавить пользователя

Отправить показания счётчиков

Уведомить о своем доходе

Отправить заявку

Получить уведомление

16

Рисунок Б.16 — Слайд презентации № 16

Заключение

В ходе выпускной квалификационной работы были решены следующие задачи:

- 1) анализ предметной области;
- 2) проектирование информационной системы с web-интерфейсом;
- 3) разработка информационной системы с web-интерфейсом.